(12)特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2001年12月27日 (27.12.2001)

PCT

(10) 国際公開番号
WO 01/99331 A1

(51) 国際特許分類[7]: H04L 9/00

(21) 国際出願番号: PCT/JP01/05146

(22) 国際出願日: 2001年6月15日 (15.06.2001)

(25) 国際出願の言語: 日本語

(26) 国際公開の言語: 日本語

(30) 優先権データ:
特願2000-179693 2000年6月15日 (15.06.2000) JP
特願2000-179694 2000年6月15日 (15.06.2000) JP

(71) 出願人 (米国を除く全ての指定国について): ソニー株式会社 (SONY CORPORATION) [JP/JP]; 〒141-0001 東京都品川区北品川6丁目7番35号 Tokyo (JP).

(72) 発明者; および
(75) 発明者/出願人 (米国についてのみ): 北谷義道 (KITAYA, Yoshimichi) [JP/JP]. 石黒隆二 (ISHIGURO, Ryuji) [JP/JP]. 大澤義知 (OSAWA, Yoshitomo) [JP/JP]. 浅野智之 (ASANO, Tomoyuki) [JP/JP]; 〒141-0001 東京都品川区北品川6丁目7番35号 ソニー株式会社内 Tokyo (JP).

(74) 代理人: 小池 晃, 外(KOIKE, Akira et al.); 〒105-0001 東京都港区虎ノ門二丁目6番4号 第11森ビル Tokyo (JP).

(81) 指定国 (国内): CA, CN, ID, IN, KR, MX, SG, US.

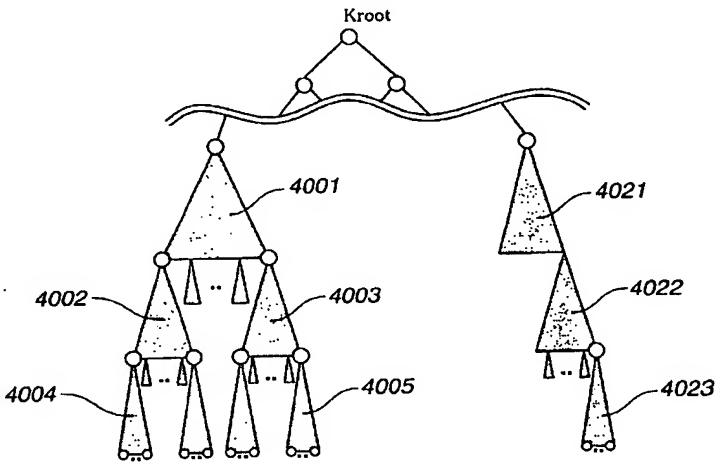
(84) 指定国 (広域): ヨーロッパ特許 (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR).

添付公開書類:
— 国際調査報告書

2文字コード及び他の略語については、定期発行される各PCTガゼットの巻頭に掲載されている「コードと略語のガイダンスノート」を参照。

(54) Title: SYSTEM AND METHOD FOR PROCESSING INFORMATION USING ENCRYPTION KEY BLOCK

(54) 発明の名称: 暗号鍵ブロックを用いた情報処理システム及び方法



(57) Abstract: Sub-trees separated according to the data processing capability of a device are defined in a key tree in which keys are assigned to the roots, nodes, and leaves which are devices of a tree. A management entity of each sub-tree creates a sub make-effective key block effective in the entity. A key issuing center creates a make-effective key block decodable only by an entity having common capability according to the capability information on the entity. Each entity manages the partia trees (sub-trees) of the key tree, creates a sub make-effective key block according to only the nodes of the sub-trees or the keys corresponding to the leaves, and creates a make-effective key block decodable only by the selected entity by using the sub make-effective key block. Thus, a make-effective key block can be created in accordance with the data processing capability of a device and distributed, and the key tree hierarchical structure can be dividedly managed.

[続葉有]

WO 01/99331 A1

(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(51) Int. Cl.		(11) 공개번호	특2002-0041804
H04L 9/08		(43) 공개일자	2002년06월03일
(21) 출원번호	10-2002-7001922		
(22) 출원일자	2002년02월14일		
번역문 제출일자	2002년02월14일		
(86) 국제출원번호	PCT/JP2001/05146		
(86) 국제출원출원일자	2001년06월15일		
(87) 국제공개번호	WO 2001/99331		
(87) 국제공개일자	2001년12월27일		
(81) 지정국	국내특허: 캐나다, 중국, 대한민국, 멕시코, 미국, 싱가포르, 인도네시아, 인도 EP 유럽특허: 오스트리아, 벨기에, 스위스, 독일, 덴마크, 스페인, 프랑스, 영국, 그리스, 아일랜드, 이탈리아, 룩셈부르크, 모나코, 네덜란드, 포르투칼, 스웨덴, 핀랜드, 사이프러스, 터어키		
(30) 우선권주장	JP-P-2000-00179693 2000년06월15일 일본(JP) JP-P-2000-00179694 2000년06월15일 일본(JP)		

(71) 출원인	소니 가부시끼 가이샤, 이데이 노부유끼 일본 000-000 일본국 도쿄도 시나가와쿠 키타시나가와 6쵸메 7반 35고
(72) 발명자	기따야,요시미찌 일본 일본141-0001도꾜도시나가와꾸기따시나가와6쪼메7-35소니가부시끼가이샤내 이시구로,류지 일본 일본141-0001도꾜도시나가와꾸기따시나가와6쪼메7-35소니가부시끼가이샤내 오사와,요시또모 일본 일본141-0001도꾜도시나가와꾸기따시나가와6쪼메7-35소니가부시끼가이샤내 아사노,도모유끼 일본 일본141-0001도꾜도시나가와꾸기따시나가와6쪼메7-35소니가부시끼가이샤내
(74) 대리인	장수길 구영창
(77) 심사청구	없음

(54) 출원명 암호 키 블록을 이용한 정보 처리 시스템 및 방법

명세서

기술분야

본 발명은 암호 키 블록을 이용한 정보 처리 시스템 및 정보 처리 방법, 및 프로그램 제공 매체에 관한 것으로, 특히 암호 처리를 수반하는 시스템에 있어서의 암호 처리 키를 배신(配信)하는 시스템 및 방법에 관한 것이다. 특히, 트리 구조의 계층적 키 배신 방식을 이용함으로써, 메시지량을 작게 억제하여, 예를 들면 콘텐츠 키 배신, 또는 각종 키의 갱신 시의 데이터 배신의 부하를 경감하고, 또한 데이터의 안전성을 유지할 수 있음과 함께, 계층적 키 배신 트리를 관리하의 디바이스의 데이터 처리 능력으로서의 캐이퍼빌리티에 기초하여 구분한 서브 트리로서의 엔티티로 관리하는 구성으로서 캐이퍼빌리티에 기초한 키 배신 및 관리 구성을 실현한 암호 키 블록을 이용한 정보 처리 시스템 및 정보 처리 방법, 및 프로그램 제공 매체, 및 계층적 키 배신 트리를 공통 요소를 갖는 부분 집합으로서의 엔티티로 관리하는 구성으로서 효율적인 키 배신 및 관리 구성을 실현한 암호 키 블록을 이용한 정보 처리 시스템 및 정보 처리 방법, 및 프로그램 제공 매체에 관한 것이다.

배경기술

최근, 게임 프로그램, 음성 데이터, 화상 데이터 등, 여러가지 소프트웨어 데이터(이하, 이들을 콘텐츠(Content)라 함)를 인터넷 등의 네트워크, 또는 DVD, CD 등의 유통 가능한 기억 매체를 통한 유통이 성행해 왔다. 이들 유통 콘텐츠는 사용자가 소유한 PC(Personal Computer), 게임 기기에 의해 데이터 수신, 또는 기억 매체의 장착이 이루어져 재생되거나, 또는 PC 등에 부속하는 기록 재생 기기 내의 기록 디바이스, 예를 들면 메모리 카드, 하드 디스크 등에 저장되어, 저장 매체로부터의 새로운 재생에 의해 이용된다.

비디오 게임 기기, PC 등의 정보 기기에는 유통 콘텐츠를 네트워크로부터 수신하기 위한, 또는 DVD, CD 등에 액세스하기 위한 인터페이스를 갖고, 또한 콘텐츠 재생에 필요한 제어 수단, 프로그램, 데이터의 메모리 영역으로서 사용되는 RAM, ROM 등을 갖는다.

음악 데이터, 화상 데이터, 또는 프로그램 등의 여러가지 콘텐츠는 재생 기기로서 이용되는 게임 기기, PC 등의 정보 기기 본체로부터의 사용자 지시, 또는 접속된 입력 수단을 통한 사용자 지시에 의해 기억 매체로부터 호출되고, 정보 기기 본체, 또는 접속된 디스플레이, 스피커 등을 통해 재생된다.

게임 프로그램, 음악 데이터, 화상 데이터 등, 많은 소프트웨어·콘텐츠는 일반적으로 그 작성자, 판매자에게 반포권 등이 보유되어 있다. 따라서, 이들 콘텐츠의 배포에 있어서는 일정한 이용 제한, 즉, 정규 사용자에게만 소프트웨어의 사용을 허락하고, 무허가 복제 등이 행해지지 않도록 하는, 즉 시큐러티를 고려한 구성을 취하는 것이 일반적이다.

사용자에 대한 이용 제한을 실현하는 하나의 방법이 배포 콘텐츠의 암호화 처리이다. 즉, 예를 들면 인터넷 등을 통해 암호화된 음성 데이터, 화상 데이터, 게임 프로그램 등의 각종 콘텐츠를 배포함과 함께, 정규 사용자라고 확인된 자에게만, 배포된 암호화 콘텐츠를 복호하는 수단, 즉 복호 키를 부여하는 구성이다.

암호화 데이터는 소정의 수속에 의한 복호화 처리에 의해 이용 가능한 복호 데이터(평문)로 복귀할 수 있다. 이러한 정보의 암호화 처리에 암호화 키를 이용하고, 복호화 처리에 복호화 키를 이용하는 데이터 암호화, 복호화 방법은 종래부터 잘 알려져 있다.

암호화 키와 복호화 키를 이용하는 데이터 암호화·복호화 방법의 양태에는 여러가지 종류가 있지만, 그 하나의 예로서, 소위 공통 키 암호화 방식이라 불리는 방식이 있다. 공통 키 암호화 방식은 데이터의 암호화 처리에 이용하는 암호화 키와 데이터의 복호화에 이용하는 복호화 키를 공통의 것으로 하고, 정규 사용자에게 이들 암호화 처리, 복호화에 이용하는 공통 키를 부여하여, 키를 갖지 않은 부정 사용자에 의한 데이터 액세스를 배제하는 것이다. 이 방식의 대표적인 방식으로 DES(데이터 암호 표준: Data encryption standard)가 있다.

상술한 암호화 처리, 복호화에 이용되는 암호화 키, 복호화 키는 예를 들면 임의의 패스워드 등에 기초하여 해시 함수 등의 일방향성 함수를 적용하여 얻을 수 있다. 일방향성 함수는, 그 출력으로부터 반대로 입력을 구하는 것은 매우 곤란한 함수이다. 예를 들면, 사용자가 결정한 패스워드를 입력으로 하여 일방향성 함수를 적용하여, 그 출력에 기초하여 암호화 키, 복호화 키를 생성하는 것이다. 이와 같이 하여 얻어진 암호화 키, 복호화 키로부터, 반대로 그 오리지널 데이터인 패스워드를 구하는 것은 실질적으로 불가능하다.

또한, 암호화할 때에 사용하는 암호화 키에 의한 처리와, 복호할 때에 사용하는 복호화 키의 처리를 다른 알고리즘으로 한 방식이, 소위 공개 키 암호화 방식이라 불리는 방식이다. 공개 키 암호화 방식은, 불특정 사용자가 사용 가능한 공개 키를 사용하는 방법으로, 특정 개인에 대한 암호화 문서를 그 특정 개인이 발행한 공개 키를 이용하여 암호화 처리를 행한다. 공개 키에 의해 암호화된 문서는 그 암호화 처리에 사용된 공개 키에 대응하는 비밀 키에 의해서만 복호 처리가 가능하게 된다. 비밀 키는 공개 키를 발행한 개인만이 소유하기 때문에, 그 공개 키에 의해 암호화된 문서는 비밀 키를 갖는 개인만이 복호할 수 있다. 공개 키 암호화 방식의 대표적인 것에는 RSA(Rivest-Shamir-Adleman) 암호가 있다. 이러한 암호화 방식을 이용함으로써, 암호화 콘텐츠를 정규 사용자에게만 복호 가능하게 하는 시스템이 가능하게 된다.

상기한 바와 같은 콘텐츠 배신 시스템에서는 콘텐츠를 암호화하여 사용자에게 네트워크, 또는 DVD, CD 등의 기록 매체에 저장하여 제공하고, 암호화 콘텐츠를 복호하는 콘텐츠 키를 정당한 사용자에게만 제공하는 구성이 많이 채용되고 있다. 콘텐츠 키 자체의 부정 복사 등을 방지하기 위한 콘텐츠 키를 암호화하여 정당한 사용자에게 제공하고, 정당한 사용자만이 갖는 복호 키를 이용하여 암호화 콘텐츠 키를 복호하여 콘텐츠 키를 사용 가능하게 하는 구성이 제안되어 있다.

정당한 사용자인지의 여부의 판정은, 일반적으로는 예를 들면 콘텐츠 송신자인 콘텐츠 프로바이더와 사용자 디바이스 사이에서, 콘텐츠, 또는 콘텐츠 키의 배신 전에 인증 처리를 실행함으로써 행한다. 일반적인 인증 처리에 있어서는 상대의 확인을 행함과 함께, 그 통신에서만 유효한 세션 키를 생성하고, 인증이 성립한 경우에, 생성한 세션 키를 이용하여 데이터, 예를 들면 콘텐츠 또는 콘텐츠 키를 암호화하여 통신을 행한다. 인증 방식에는 공통 키 암호 방식을 이용한 상호 인증과, 공개 키 방식을 사용한 인증 방식이 있지만, 공통 키를 사용한 인증에 있어서는 시스템 와이드로 공통의 키가 필요하게 되고, 갱신 처리 등을 할 때에 불편하다. 또한, 공개 키 방식에 있어서는 계산 부하가 크고, 또한 필요한 메모리량도 커져, 각 디바이스에 이러한 처리 수단을 설치하는 것은 바람직한 구성이라고는 할 수 없다.

〈발명의 개시〉

본 발명의 목적은 상술한 바와 같은 데이터의 송신자, 수신자간의 상호 인증 처리에 의지하지 않고 정당한 사용자에게만, 안전하게 데이터를 송신할 수 있음과 함께, 계층적 키 배신 트리를 관리 하의 디바이스의 데이터 처리 능력으로서의 캐이퍼빌리티에 기초하여 구분한 서브 트리로서의 엔티티로 관리하는 구성으로서 캐이퍼빌리티에 기초한 키 배신 및 관리 구성을 실현한 암호 키 블록을 이용한 정보 처리 시스템 및 정보 처리 방법, 및 프로그램 제공 매체를 제공하는 것이다.

또한, 본 발명의 목적은 상술한 바와 같은 데이터의 송신자, 수신자간의 상호 인증 처리에 의지하지 않고 정당한 사용자에게만, 안전하게 데이터를 송신할 수 있음과 함께, 계층적 키 배신 트리를 공통 요소를 갖는 부분 집합으로서의 엔티티로 관리하는 구성으로서 효율적인 키 배신 및 관리 구성을 실현한 암호 키 블록을 이용한 정보 처리 시스템 및 정보 처리 방법, 및 프로그램 제공 매체를 제공하는 것이다.

본 발명에 따른 암호 키 블록을 이용한 정보 처리 시스템은 복수의 디바이스를 리프로 구성한 트리의 루트에서 리프까지의 패스 상의 루트, 노드, 및 리프에 각각 키를 대응시킨 키 트리를 구성하고, 키 트리를 구성하는 패스를 선택하여 선택 패스 상의 키 갱신, 및 하위 키에 의한 상위 키의 암호화 처리를 실행하여 특정 디바이스에 있어서만 복호 가능한 유효화 키 블록(EKB)을 생성하여 디바이스에 제공하는 암호 키 블록을 이용한 정보 처리 시스템에 있어서, 키 트리의 일부를 구성하고, 디바이스의 데이터 처리 능력으로서의 캐이퍼빌리티에 기초하여 구분된 서브 트리를 관리하고, 서브 트리에 속하는 노드 또는 리프에 대응하여 설정되는 키에만 기초한 서브 유효화 키 블록(서브 EKB)을 생성하는 복수의 엔티티와, 복수의 엔티티의 캐이퍼빌리티 정보를 관리하고, 공통의 캐이퍼빌리티를 갖는 엔티티가 생성하는 서브 유효화 키 블록(서브 EKB)을 이용하여, 공통의 캐이퍼빌리티를 갖는 엔티티에 있어서만 복호 가능한 유효화 키 블록(EKB)을 생성하는 키 발행 센터(KDC)를 포함한다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 시스템에 있어서, 키 발행 센터(KDC)는 복수의 엔티티 각각의 식별자와, 엔티티 각각의 캐이퍼빌리티 정보와, 엔티티 각각의 서브 유효화 키 블록(서브 EKB) 정보를 대응시킨 캐이퍼빌리티 관리 테이블을 갖고, 캐이퍼빌리티 관리 테이블에 기초하여 디바이스에 대한 배신 데이터의 처리 가능한 엔티티를 선택하여, 선택 엔티티 산하의 디바이스에서만 복호 가능한 유효화 키 블록(EKB)을 생성하는 구성을 갖는다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 시스템에 있어서, 키 트리에 대한 신규 추가 엔티티는 신규 엔티티 내의 서브 트리 내의 노드 또는 리프에 대응하여 설정되는 키에만 기초한 서브 유효화 키 블록(서브 EKB)을 생성하고, 키 발행 센터(KDC)에 대한 서브 EKB의 등록 처리를 실행함과 함께, 자신의 엔티티의 캐이퍼빌리티 정보의 통지 처리를 실행하는 구성이다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 시스템에 있어서, 복수의 엔티티는 하나의 엔티티의 최하단의 말단 노드를 다른 엔티티의 정점 노드(서브 루트)로서 구성한 상위 엔티티 및 하위 엔티티의 계층화 구조를 갖는다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 시스템에 있어서, 복수의 엔티티의 각각은 자신의 엔티티에 속하는 서브 트리를 구성하는 노드 또는 리프에 대응하는 키의 설정, 갱신 처리 권한을 갖는 구성이다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 시스템에 있어서, 복수의 엔티티 중, 엔티티 내의 최하단 리프를 개개의 디바이스에 대응하는 리프로 한 최하층의 엔티티에 속하는 디바이스의 각각은 자신이 속하는 엔티티의 정점 노드(서브 루트)로부터 자신의 디바이스에 대응하는 리프에 이르는 패스 상의 노드, 리프에 설정된 노드 키 및 리프 키를 저장한 구성을 갖는다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 시스템에 있어서, 복수의 엔티티의 각각은 자신의 엔티티의 하위에, 또한 자기 관리 엔티티를 추가하기 위해서, 자신의 엔티티 내의 최하단의 노드 또는 리프 중의 1이상의 노드 또는 리프를 리저브(reserved) 노드로서 보류하여 설정한 구성을 갖는다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 시스템에 있어서, 신규 엔티티를 말단 노드에 추가하는 상위 엔티티는 신규 엔티티의 서브 트리를 설정하는 노드인 상위 엔티티 말단 노드에 대응하는 키를 신규 엔티티의 정점 노드(서브 루트) 키로서 설정하는 구성이다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 시스템에 있어서, 디바이스의 리보크(revoke) 처리를 실행하는 엔티티는 엔티티 내의 정점 노드(서브 루트)로부터 리보크 디바이스에 대응하는 리프에 이르는 패스 상의 노드에 설정된 노드 키를 갱신하고, 갱신 노드 키를 리보크 디바이스 이외의 리프 디바이스에 있어서만 복호 가능한 암호화 키로서 구성한 갱신 서브 EKB를 생성하여 상위 엔티티에 송신하고, 상위 엔티티는 갱신 서브 EKB를 제공한 말단 노드로부터 자신의 서브 루트에 이르는 패스 상의 노드 키를 갱신한 갱신 서브 EKB를 생성하여 다시 상위 엔티티에 송신하고, 최상위 엔티티까지, 엔티티 단위의 갱신 서브 EKB 생성 및 송신 처리를 순차적으로 실행하여 리보크 디바이스로부터 루트에 이르는 패스 상의 노드 키 갱신을 행하고, 키 갱신에 의해 생성된 갱신 서브 EKB의 키 발행 센터(KDC)에의 등록 처리를 행함으로써, 디바이스의 리보크 처리를 실행하는 구성을 갖는다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 시스템에 있어서, 하위 엔티티의 리보크 처리를 실행하는 엔티티는 엔티티 내의 정점 노드(서브 루트)로부터 리보크·엔티티에 대응하는 말단 노드에 이르는 패스 상의 노드에 설정된 노드 키를 갱신한 갱신 서브 EKB를 생성하여 상위 엔티티에 송신하고, 상위 엔티티는 갱신 서브 EKB를 제공한 말단 노드로부터 자신의 서브 루트에 이르는 패스 상의 노드 키를 갱신한 갱신 서브 EKB를 생성하여 다시 상위 엔티티에 송신하고, 최상위 엔티티까지, 엔티티 단위의 갱신 서브 EKB 생성 및 송신 처리를 순차적으로 실행하여 리보크·엔티티로부터 루트에 이르는 패스 상의 노드 키 갱신을 행하고, 키 갱신에 의해 생성된 갱신 서브 EKB의 키 발행 센터(KDC)에의 등록 처리를 행함으로써, 엔티티 단위의 리보크 처리를 실행하는 구성을 갖는다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 시스템에 있어서, 하위 엔티티의 리보크 처리를 실행하는 엔티티는 엔티티 내의 정점 노드(서브 루트)로부터 리보크·엔티티에 대응하는 말단 노드에 이르는 패스 상의, 말단 노드를 제외한 노드에 설정된 노드 키를 갱신한 갱신 서브 EKB를 생성하여 상위 엔티티에 송신하고, 상위 엔티티는 갱신 서브 EKB를 제공한 말단 노드로부터 자신의 서브 루트에 이르는 패스 상의 노드 키를 갱신한 갱신 서브 EKB를 생성하여 다시 상위 엔티티에 송신하고, 최상위 엔티티까지, 엔티티 단위의 갱신 서브 EKB 생성 및 송신 처리를 순차적으로 실행하여 리보크·엔티티로부터 루트에 이르는 패스 상의 리보크·엔티티에 대응하는 말단 노드를 제외한 노드 키 갱신을 행하고, 키 갱신에 의해 생성된 갱신 서브 EKB의 키 발행 센터(KDC)에의 등록 처리를 행함으로써, 엔티티 단위의 리보크 처리를 실행하는 구성을 갖는다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 방법은 복수의 디바이스를 리프로 구성한 트리의 루트에서 리프까지의 패스 상의 루트, 노드, 및 리프에 각각 키를 대응시킨 키 트리를 구성하고, 키 트리를 구성하는 패스를 선택하여 선택 패스 상의 키 갱신, 및 하위 키에 의한 상위 키의 암호화 처리를 실행하여 특정 디바이스에 있어서만 복호 가능한 유효화 키 블록(EKB)을 생성하여 디바이스에 제공하는 정보 처리 시스템에 있어서의 암호 키 블록을 이용한 정보 처리 방법에 있어서, 키 트리의 일부를 구성하고, 디바이스의 데이터 처리 능력으로서의 캐이퍼빌리티에 기초하여 구분된 서브 트리를 관리하는 엔티티에 있어서, 각 엔티티의 서브 트리에 속하는 노드 또는 리프에 대응하여 설정되는 키에만 기초한 서브 유효화 키 블록(서브 EKB)을 생성하는 단계와, 복수의 엔티티의 캐이퍼빌리티 정보를 보유하는 키 발행 센터(KDC)에서, 복수의 엔티티의 캐이퍼빌리티 정보에 기초하여 공통의 캐이퍼빌리티를 갖는 엔티티가 생성하는 서브 유효화 키 블록(서브 EKB)을 추출하여 공통의 캐이퍼빌리티를 갖는 엔티티에 있어서만 복호 가능한 유효화 키 블록(EKB)을 생성하는 단계를 포함한다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 방법에 있어서, 키 발행 센터(KDC)에 있어서의 유효화 키 블록(EKB) 생성 단계는 공통의 캐이퍼빌리티를 갖는 엔티티를 선택하는 엔티티 선택 단계와, 엔티티 선택 단계에서 선택된 엔티티에 의해 구성되는 엔티티·트리를 생성하는 단계와, 엔티티·트리를 구성하는 노드 키를 갱신하는 노드 키 갱신 단계와, 노드 키 갱신 단계에서 갱신한 노드 키, 및 선택 엔티티의 서브 EKB에 기초하여 선택 엔티티에 있어서만 복호 가능한 유효화 키 블록(EKB)을 생성하는 단계를 포함한다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 방법에 있어서, 키 발행 센터(KDC)는 복수의 엔티티 각각의 식별자와, 엔티티 각각의 캐이퍼빌리티 정보와, 엔티티 각각의 서브 유효화 키 블록(서브 EKB) 정보를 대응시킨 캐이퍼빌리티 관리 테이블을 갖고, 캐이퍼빌리티 관리 테이블에 기초하여 디바이스에 대한 배신 데이터의 처리 가능한 엔티티를 선택하여, 선택 엔티티 산하의 디바이스에서만 복호 가능한 유효화 키 블록(EKB)을 생성한다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 방법에 있어서, 키 트리에 대한 신규 추가 엔티티는 신규 엔티티 내의 서브 트리 내의 노드 또는 리프에 대응하여 설정되는 키에만 기초한 서브 유효화 키 블록(서브 EKB)을 생성하고, 키 발행 센터(KDC)에 대한 서브 EKB의 등록 처리를 실행함과 함께, 자신의 엔티티의 캐이퍼빌리티 정보의 통지 처리를 실행한다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 방법에 있어서, 복수의 엔티티의 각각은 자신의 엔티티에 속하는 서브 트리를 구성하는 노드 또는 리프에 대응하는 키의 설정, 갱신 처리를 실행한다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 방법에 있어서, 신규 엔티티를 말단 노드에 추가하는 상위 엔티티는 신규 엔티티의 서브 트리를 설정하는 노드인 상위 엔티티 말단 노드에 대응하는 키를 신규 엔티티의 정점 노드(서브 루트) 키로서 설정한다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 방법에 있어서, 디바이스의 리보크 처리를 실행하는 엔티티는 엔티티 내의 정점 노드(서브 루트)로부터 리보크 디바이스에 대응하는 리프에 이르는 패스 상의 노드에 설정된 노드 키를 갱신하고, 갱신 노드 키를 리보크 디바이스 이외의 리프 디바이스에 있어서만 복호

가능한 암호화 키로서 구성한 갱신 서브 EKB를 생성하여 상위 엔티티에 송신하고, 상위 엔티티는 갱신 서브 EKB를 제공한 말단 노드로부터 자신의 서브 루트에 이르는 패스 상의 노드 키를 갱신한 갱신 서브 EKB를 생성하여 다시 상위 엔티티에 송신하고, 최상위 엔티티까지, 엔티티 단위의 갱신 서브 EKB 생성 및 송신 처리를 순차적으로 실행하여 리보크 디바이스로부터 루트에 이르는 패스 상의 노드 키 갱신을 행하고, 키 갱신에 의해 생성된 갱신 서브 EKB의 키 발행 센터(KDC)에의 등록 처리를 행함으로써, 디바이스의 리보크 처리를 실행한다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 방법에 있어서, 하위 엔티티의 리보크 처리를 실행하는 엔티티는 엔티티 내의 정점 노드(서브 루트)로부터 리보크·엔티티에 대응하는 말단 노드에 이르는 패스 상의 노드에 설정된 노드 키를 갱신한 갱신 서브 EKB를 생성하여 상위 엔티티에 송신하고, 상위 엔티티는 갱신 서브 EKB를 제공한 말단 노드로부터 자신의 서브 루트에 이르는 패스 상의 노드 키를 갱신한 갱신 서브 EKB를 생성하여 다시 상위 엔티티에 송신하고, 최상위 엔티티까지, 엔티티 단위의 갱신 서브 EKB 생성 및 송신 처리를 순차적으로 실행하여 리보크·엔티티로부터 루트에 이르는 패스 상의 노드 키 갱신을 행하고, 키 갱신에 의해 생성된 갱신 서브 EKB의 키 발행 센터(KDC)에의 등록 처리를 행함으로써, 엔티티 단위의 리보크 처리를 실행한다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 방법에 있어서, 하위 엔티티의 리보크 처리를 실행하는 엔티티는 엔티티 내의 정점 노드(서브 루트)로부터 리보크·엔티티에 대응하는 말단 노드에 이르는 패스 상의 말단 노드를 제외한 노드에 설정된 노드 키를 갱신한 갱신 서브 EKB를 생성하여 상위 엔티티에 송신하고, 상위 엔티티는 갱신 서브 EKB를 제공한 말단 노드로부터 자신의 서브 루트에 이르는 패스 상의 노드 키를 갱신한 갱신 서브 EKB를 생성하여 다시 상위 엔티티에 송신하고, 최상위 엔티티까지, 엔티티 단위의 갱신 서브 EKB 생성 및 송신 처리를 순차적으로 실행하여 리보크·엔티티로부터 루트에 이르는 패스 상의 리보크·엔티티에 대응하는 말단 노드를 제외한 노드 키 갱신을 행하고, 키 갱신에 의해 생성된 갱신 서브 EKB의 키 발행 센터(KDC)에의 등록 처리를 행함으로써, 엔티티 단위의 리보크 처리를 실행한다.

또한, 본 발명에 따른 프로그램 제공 매체는 복수의 디바이스를 리프로 구성한 트리의 루트에서 리프까지의 패스 상의 루트, 노드, 및 리프에 각각 키를 대응시킨 키 트리를 구성하고, 키 트리를 구성하는 패스를 선택하여 선택 패스 상의 키 갱신, 및 하위 키에 의한 상위 키의 암호화 처리를 실행하여 특정 디바이스에 있어서만 복호 가능한 유효화 키 블록(EKB)을 생성하여 디바이스에 제공하는 정보 처리 시스템에 있어서의 유효화 키 블록(EKB) 생성 처리를 컴퓨터 시스템 상에서 실행시키는 컴퓨터 프로그램을 제공한다. 컴퓨터 프로그램은 키 트리 일부를 구성하고, 디바이스의 데이터 처리 능력으로서의 캐이퍼빌리티에 기초하여 구분된 서브 트리를 관리하는 엔티티에 있어서, 각 엔티티의 서브 트리에 속하는 노드 또는 리프에 대응하여 설정되는 키에만 기초한 서브 유효화 키 블록(서브 EKB)을 생성하는 단계와, 복수의 엔티티의 캐이퍼빌리티 정보를 보유하는 키 발행 센터(KDC)에서 복수의 엔티티의 캐이퍼빌리티 정보에 기초하여 공통의 캐이퍼빌리티를 갖는 엔티티가 생성하는 서브 유효화 키 블록(서브 EKB)을 추출하여 공통의 캐이퍼빌리티를 갖는 엔티티에 있어서만 복호 가능한 유효화 키 블록(EKB)을 생성하는 단계를 포함한다.

본 발명에 따른 암호 키 블록을 이용한 정보 처리 시스템은 복수의 디바이스를 리프로 구성한 트리의 루트에서 리프까지의 패스 상의 루트, 노드, 및 리프에 각각 키를 대응시킨 키 트리를 구성하고, 키 트리를 구성하는 패스를 선택하여 선택 패스 상의 키 갱신, 및 하위 키에 의한 상위 키의 암호화 처리를 실행하여 특정 디바이스에 있어서만 복호 가능한 유효화 키 블록(EKB)을 생성하여 디바이스에 제공하는 암호 키 블록을 이용한 정보 처리 시스템에 있어서, 키 트리를 구성하는 부분 트리로서의 서브 트리를 관리하고, 서브 트리에 속하는 노드 또는 리프에 대응하여 설정되는 키에만 기초한 서브 유효화 키 블록(서브 EKB)을 생성하는 복수의 엔티티와, 복수의 엔티티가 생성하는 서브 유효화 키 블록(서브 EKB)을 이용하여, 선택된 엔티티에 있어서만 복호 가능한 유효화 키 블록(EKB)을 생성하는 키 발행 센터(KDC)를 갖는다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 시스템에 있어서, 복수의 엔티티는 하나의 엔티티의 최하단의 말단 노드를 다른 엔티티의 정점 노드(서브 루트)로서 구성한 상위 엔티티 및 하위 엔티티의 계층화 구조를 갖는다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 시스템에 있어서, 복수의 엔티티의 각각은 자신의 엔티티에 속하는 서브 트리를 구성하는 노드 또는 리프에 대응하는 키의 설정, 갱신 처리 권한을 갖는 구성이다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 시스템에 있어서, 복수의 엔티티 중, 엔티티 내의 최하단 리프를 개개의 디바이스에 대응하는 리프로 한 최하층의 엔티티에 속하는 디바이스의 각각은 자신이 속하는 엔티티의 정점 노드(서브 루트)로부터 자신의 디바이스에 대응하는 리프에 이르는 패스 상의 노드, 리프에 설정된 노드 키 및 리프 키를 저장한 구성을 갖는다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 시스템에 있어서, 복수의 엔티티의 각각은 자신의 엔티티의 하위에, 또한 자기 관리 엔티티를 추가하기 위해서, 자신의 엔티티 내의 최하단의 노드 또는 리프 중의 1이상의 노드 또는 리프를 리저브 노드로서 보류하여 설정한 구성을 갖는다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 시스템에 있어서, 신규 엔티티를 말단 노드에 추가하는 상위 엔티티는 신규 엔티티의 서브 트리를 설정하는 노드인 상위 엔티티 말단 노드에 대응하는 키를 신규 엔티티의 정점 노드(서브 루트) 키로서 설정하는 구성이다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 시스템에 있어서, 신규 추가 엔티티는 신규 엔티티 내의 서브 트리 내의 노드 또는 리프에 대응하여 설정되는 키에만 기초한 서브 유효화 키 블록(서브 EKB)을 생성하고, 키 발행 센터(KDC)에 대한 서브 EKB의 등록 처리를 실행하는 구성이다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 시스템에 있어서, 디바이스의 리보크 처리를 실행하는 엔티티는 엔티티 내의 정점 노드(서브 루트)로부터 리보크 디바이스에 대응하는 리프에 이르는 패스 상의 노드에 설정된 노드 키를 갱신하고, 갱신 노드 키를 리보크 디바이스 이외의 리프 디바이스에 있어서만 복호 가능한 암호화 키로서 구성한 갱신 서브 EKB를 생성하여 상위 엔티티에 송신하고, 상위 엔티티는 갱신 서브 EKB를 제공한 말단 노드로부터 자신의 서브 루트에 이르는 패스 상의 노드 키를 갱신한 갱신 서브 EKB를 생성하여 다시 상위 엔티티에 송신하고, 최상위 엔티티까지, 엔티티 단위의 갱신 서브 EKB 생성 및 송신 처리를 순차적으로 실행하여 리보크 디바이스로부터 루트에 이르는 패스 상의 노드 키 갱신을 행하고, 키 갱신에 의해 생성된 갱신 서브 EKB의 키 발행 센터(KDC)에의 등록 처리를 행함으로써, 디바이스의 리보크 처리를 실행하는 구성을 갖는다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 시스템에 있어서, 하위 엔티티의 리보크 처리를 실행하는 엔티티는 엔티티 내의 정점 노드(서브 루트)로부터 리보크·엔티티에 대응하는 말단 노드에 이르는 패스 상의 노드에 설정된 노드 키를 갱신한 갱신 서브 EKB를 생성하여 상위 엔티티에 송신하고, 상위 엔티티는 갱신 서브 EKB를 제공한 말단 노드로부터 자신의 서브 루트에 이르는 패스 상의 노드 키를 갱신한 갱신 서브 EKB를 생성하여 다시 상위 엔티티에 송신하고, 최상위 엔티티까지, 엔티티 단위의 갱신 서브 EKB 생성 및 송신 처리를 순차적으로 실행하여 리보크·엔티티로부터 루트에 이르는 패스 상의 노드 키 갱신을 행하고, 키 갱신에 의해 생성된 갱신 서브 EKB의 키 발행 센터(KDC)에의 등록 처리를 행함으로써, 엔티티 단위의 리보크 처리를 실행하는 구성을 갖는다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 시스템에 있어서, 하위 엔티티의 리보크 처리를 실행하는 엔티티는 엔티티 내의 정점 노드(서브 루트)로부터 리보크·엔티티에 대응하는 말단 노드에 이르는 패스 상의, 말단 노드를 제외한 노드에 설정된 노드 키를 갱신한 갱신 서브 EKB를 생성하여 상위 엔티티에 송신하고, 상위 엔티티는 갱신 서브 EKB를 제공한 말단 노드로부터 자신의 서브 루트에 이르는 패스 상의 노드 키를 갱신한 갱신 서브 EKB를 생성하여 다시 상위 엔티티에 송신하고, 최상위 엔티티까지, 엔티티 단위의 갱신 서브 EKB 생성 및 송신 처리를 순차적으로 실행하여 리보크·엔티티로부터 루트에 이르는 패스 상의 리보크·엔티티에 대응하는 말단 노드를 제외한 노드 키 갱신을 행하고, 키 갱신에 의해 생성된 갱신 서브 EKB의 키 발행 센터(KDC)에의 등록 처리를 행함으로써, 엔티티 단위의 리보크 처리를 실행하는 구성을 갖는다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 시스템에 있어서, 엔티티는 디바이스 종류, 서비스 종류, 관리 수단 종류 등의 공통의 카테고리에 속하는 디바이스 또는 엔티티의 관리 주체로서 구성된다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 방법은 복수의 디바이스를 리프로 구성한 트리의 루트에서 리프까지의 패스 상의 루트, 노드, 및 리프에 각각 키를 대응시킨 키 트리를 구성하고, 키 트리를 구성하는 패스를 선택하여 선택 패스 상의 키 갱신, 및 하위 키에 의한 상위 키의 암호화 처리를 실행하여 특정 디바이스에 있어서만 복호 가능한 유효화 키 블록(EKB)을 생성하여 디바이스에 제공하는 정보 처리 시스템에 있어서의 암호 키 블록을 이용한 정보 처리 방법에 있어서, 키 트리를 구성하는 부분 트리로서의 서브 트리를 관리하는 복수의 엔티티에 있어서, 서브 트리에 속하는 노드 또는 리프에 대응하여 설정되는 키에만 기초한 서브 유효화 키 블록(서브 EKB)을 생성하는 단계와, 키 발행 센터(KDC)에서 복수의 엔티티가 생성하는 서브 유효화 키 블록(서브 EKB)을 이용하여, 선택된 엔티티에 있어서만 복호 가능한 유효화 키 블록(EKB)을 생성하는 단계를 포함한다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 방법에 있어서, 복수의 엔티티의 각각은 자신의 엔티티에 속하는 서브 트리를 구성하는 노드 또는 리프에 대응하는 키의 설정, 갱신 처리를 실행한다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 방법에 있어서, 신규 엔티티를 말단 노드에 추가하는 상위 엔티티는 신규 엔티티의 서브 트리를 설정하는 노드인 상위 엔티티 말단 노드에 대응하는 키를 신규 엔티티의 정점 노드(서브 루트) 키로서 설정한다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 방법에 있어서, 신규 추가 엔티티는 신규 엔티티 내의 서브 트리 내의 노드 또는 리프에 대응하여 설정되는 키에만 기초한 서브 유효화 키 블록(서브 EKB)을 생성하고, 키 발행 센터(KDC)에 대한 서브 EKB의 등록 처리를 실행한다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 방법에 있어서, 디바이스의 리보크 처리를 실행하는 엔티티는 엔티티 내의 정점 노드(서브 루트)로부터 리보크 디바이스에 대응하는 리프에 이르는 패스 상의 노드에 설정된 노드 키를 갱신하고, 갱신 노드 키를 리보크 디바이스 이외의 리프 디바이스에 있어서만 복호 가능한 암호화 키로서 구성한 갱신 서브 EKB를 생성하여 상위 엔티티에 송신하고, 상위 엔티티는 갱신 서브 EKB를 제공한 말단 노드로부터 자신의 서브 루트에 이르는 패스 상의 노드 키를 갱신한 갱신 서브 EKB를 생성하여 다시 상위 엔티티에 송신하고, 최상위 엔티티까지, 엔티티 단위의 갱신 서브 EKB 생성 및 송신 처리를 순차적으로 실행하여 리보크 디바이스로부터 루트에 이르는 패스 상의 노드 키 갱신을 행하고, 키 갱신에 의해 생성된 갱신 서브 EKB의 키 발행 센터(KDC)에의 등록 처리를 행함으로써, 디바이스의 리보크 처리를 실행한다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 방법에 있어서, 하위 엔티티의 리보크 처리를 실행하는 엔티티는 엔티티 내의 정점 노드(서브 루트)로부터 리보크·엔티티에 대응하는 말단 노드에 이르는 패스 상의 노드에 설정된 노드 키를 갱신한 갱신 서브 EKB를 생성하여 상위 엔티티에 송신하고, 상위 엔티티는 갱신 서브 EKB를 제공한 말단 노드로부터 자신의 서브 루트에 이르는 패스 상의 노드 키를 갱신한 갱신 서브 EKB를 생성하여 다시 상위 엔티티에 송신하고, 최상위 엔티티까지, 엔티티 단위의 갱신 서브 EKB 생성 및 송신 처리를 순차적으로 실행하여 리보크·엔티티로부터 루트에 이르는 패스 상의 노드 키 갱신을 행하고, 키 갱신에 의해 생성된 갱신 서브 EKB의 키 발행 센터(KDC)에의 등록 처리를 행함으로써, 엔티티 단위의 리보크 처리를 실행한다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 방법에 있어서, 하위 엔티티의 리보크 처리를 실행하는 엔티티는 엔티티 내의 정점 노드(서브 루트)로부터 리보크·엔티티에 대응하는 말단 노드에 이르는 패스 상의, 말단 노드를 제외한 노드에 설정된 노드 키를 갱신한 갱신 서브 EKB를 생성하여 상위 엔티티에 송신하고, 상위 엔티티는 갱신 서브 EKB를 제공한 말단 노드로부터 자신의 서브 루트에 이르는 패스 상의 노드 키를 갱신한 갱신 서브 EKB를 생성하여 다시 상위 엔티티에 송신하고, 최상위 엔티티까지, 엔티티 단위의 갱신 서브 EKB 생성 및 송신 처리를 순차적으로 실행하여 리보크·엔티티로부터 루트에 이르는 패스 상의 리보크·엔티티에 대응하는 말단 노드를 제외한 노드 키 갱신을 행하고, 키 갱신에 의해 생성된 갱신 서브 EKB의 키 발행 센터(KDC)에의 등록 처리를 행함으로써, 엔티티 단위의 리보크 처리를 실행한다.

또한, 본 발명에 따른 프로그램 제공 매체는 복수의 디바이스를 리프로 구성한 트리의 루트에서 리프까지의 패스 상의 루트, 노드, 및 리프에 각각 키를 대응시킨 키 트리를 구성하고, 키 트리를 구성하는 패스를 선택하여 선택 패스 상의 키 갱신, 및 하위 키에 의한 상위 키의 암호화 처리를 실행하여 특정 디바이스에 있어서만 복호 가능한 유효화 키 블록(EKB)을 생성하여 디바이스에 제공하는 정보 처리 시스템에 있어서의 유효화 키 블록(EKB) 생성 처리를 컴퓨터 시스템 상에서 실행시키는 컴퓨터 프로그램을 제공한다. 컴퓨터 프로그램은 키 트리를 구성하는 부분 트리로서의 서브 트리를 관리하는 복수의 엔티티에 있어서, 서브 트리에 속하는 노드 또는 리프에 대응하여 설정되는 키에만 기초한 서브 유효화 키 블록(서브 EKB)을 생성하는 단계와, 키 발행 센터(KDC)에서, 복수의 엔티티가 생성하는 서브 유효화 키 블록(서브 EKB)을 이용하여, 선택된 엔티티에 있어서만 복호 가능한 유효화 키 블록(EKB)을 생성하는 단계를 포함한다.

본 발명에 따른 암호 키 블록을 이용한 정보 처리 시스템 및 정보 처리 방법에서는 트리(나무) 구조의 계층적 구조의 암호화 키 배신 구성을 이용함으로써, 키 갱신에 필요한 배신 메시지량을 작게 억제하고 있다. 즉, 각 기기를 n분 나무의 각 잎(리프)에 배치한 구성의 키 배신 방법을 이용하고, 기록 매체 또는 통신 회선을 통해, 예를 들면 콘텐츠 데이터의 암호 키인 콘텐츠 키 또는 인증 처리에 이용하는 인증 키, 또는 프로그램 코드 등을 유효화 키 블록과 함께 배신하는 구성으로 하고 있다. 이와 같이 함으로써, 정당한 디바이스만이 복호 가능한 데이터를 안전하게 배신할 수 있다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 시스템 및 방법에서는 계층적 키 배신 트리를, 관리하의 디바이스의 데이터 처리 능력으로서의 캐이퍼빌리티에 기초하여 구분한 서브 트리로서의 엔티티로 관리하는 구성으로서 캐이퍼빌리티에 기초한 키 배신 및 관리 구성을 실현하고 있다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 시스템 및 방법에서는 계층적 키 배신 트리를 공통 요소를 갖는 부분 집합으로서의 엔티티로 관리하는 구성으로서 효율적인 키 배신 및 관리 구성을 실현하고 있다.

또, 본 발명에 따른 프로그램 제공 매체는 예를 들면, 여러가지 프로그램·코드를 실행할 수 있는 범용 컴퓨터 시스템에 대하여, 컴퓨터 프로그램을 컴퓨터 판독 가능한 형식으로 제공하는 매체이다. 매체는 CD나 FD, MO 등의 기록 매체, 또는 네트워크 등의 전송 매체 등, 그 형태는 특별히 한정되지 않는다.

이러한 프로그램 제공 매체는 컴퓨터 시스템 상에서 소정의 컴퓨터 프로그램의 기능을 실현하기 위한, 컴퓨터 프로그램과 제공 매체와의 구조상 또는 기능 상의 협동적 관계를 정의한 것이다. 다시 말하면, 제공 매체를 통해 컴퓨터 프로그램을 컴퓨터 시스템에 인스톨함으로써, 컴퓨터 시스템 상에서는 협동적 작용이 발휘되고, 본 발명의 다른 측면과 마찬가지의 작용 효과를 얻을 수 있는 것이다.

본 발명의 또 다른 목적, 특징이나 이점은 후술하는 본 발명의 실시예나 첨부하는 도면에 기초하여 상세한 설명에 의해 보다 분명히 될 것이다.

도면의 간단한 설명

도 1은 본 발명의 정보 처리 시스템의 구성예의 설명도.

도 2는 본 발명의 정보 처리 시스템에 있어서 적용 가능한 기록 재생 장치의 구성예를 나타내는 블록도.

도 3은 본 발명의 정보 처리 시스템에 있어서의 각종 키, 데이터의 암호화 처리에 대하여 설명하는 트리 구성도.

도 4A 및 도 4B는 본 발명의 정보 처리 시스템에 있어서의 각종 키, 데이터의 배포에 사용되는 유효화 키 블록(EKB)의 예를 나타내는 도면.

도 5는 본 발명의 정보 처리 시스템에 있어서의 콘텐츠 키의 유효화 키 블록(EKB)을 사용한 배포예와 복호 처리예를 나타내는 도면.

도 6은 본 발명의 정보 처리 시스템에 있어서의 유효화 키 블록(EKB)의 포맷예를 나타내는 도면.

도 7의 (A) 내지 도 7의 (C)는 본 발명의 정보 처리 시스템에 있어서의 유효화 키 블록(EKB)의 태그의 구성의 설명도.

도 8A 및 도 8B는 본 발명의 정보 처리 시스템에 있어서의 유효화 키 블록(EKB)과, 콘텐츠 키, 콘텐츠를 함께 배신하는 데이터 구성예를 나타내는 도면.

도 9는 본 발명의 정보 처리 시스템에 있어서의 유효화 키 블록(EKB)과, 콘텐츠 키, 콘텐츠를 함께 배신한 경우의 디바이스에서의 처리예를 나타내는 도면.

도 10은 본 발명의 정보 처리 시스템에 있어서의 유효화 키 블록(EKB)과 콘텐츠를 기록 매체에 저장한 경우의 대응에 대한 설명도.

도 11A 및 도 11B는 본 발명의 정보 처리 시스템에 있어서의 유효화 키 블록(EKB)과, 콘텐츠 키를 송부하는 처리를 종래의 송부 처리와 비교한 도면.

도 12는 본 발명의 정보 처리 시스템에 있어서 적용 가능한 공통 키 암호 방식에 의한 인증 처리 시퀀스를 나타내는 도면.

도 13은 본 발명의 정보 처리 시스템에 있어서의 유효화 키 블록(EKB)과, 인증 키를 함께 배신하는 데이터 구성과, 디바이스에서의 처리예를 나타내는 제1도.

도 14는 본 발명의 정보 처리 시스템에 있어서의 유효화 키 블록(EKB)과, 인증 키를 함께 배신하는 데이터 구성과, 디바이스에서의 처리예를 나타내는 제2도.

도 15는 본 발명의 정보 처리 시스템에 있어서 적용 가능한 공개 키 암호 방식에 의한 인증 처리 시퀀스를 나타내는 도면.

도 16은 본 발명의 정보 처리 시스템에 있어서 공개 키 암호 방식에 의한 인증 처리를 이용하여 유효화 키 블록(EKB)과, 콘텐츠 키를 함께 배신하는 처리를 나타내는 도면.

도 17은 본 발명의 정보 처리 시스템에 있어서 유효화 키 블록(EKB)과, 암호화 프로그램 데이터를 함께 배신하는 처리를 나타내는 도면.

도 18은 본 발명의 정보 처리 시스템에 있어서 적용 가능한 콘텐츠·인터그리티·체크치(ICV)의 생성에 사용하는 MAC치 생성예를 나타내는 도면.

도 19는 본 발명의 정보 처리 시스템에 있어서의 유효화 키 블록(EKB)과, ICV 생성 키를 함께 배신하는 데이터 구성과, 디바이스에서의 처리예를 나타내는 제1도.

도 20은 본 발명의 정보 처리 시스템에 있어서의 유효화 키 블록(EKB)과, ICV 생성 키를 함께 배신하는 데이터 구성과, 디바이스에서의 처리예를 나타내는 제2도.

도 21A 및 도 21B는 본 발명의 정보 처리 시스템에 있어서 적용 가능한 콘텐츠·인터그리티·체크치(ICV)를 미디어에 저장한 경우의 복사 방지 기능의 설명도.

도 22는 본 발명의 정보 처리 시스템에 있어서 적용 가능한 콘텐츠·인터그리티·체크치(ICV)를 콘텐츠 저장 매체와 별도로 관리하는 구성의 설명도.

도 23은 본 발명의 정보 처리 시스템에 있어서의 계층 트리 구조의 카테고리 분류예의 설명도.

도 24A 및 도 24B는 본 발명의 정보 처리 시스템에 있어서의 간략화 유효화 키 블록(EKB)의 생성 과정의 설명도.

도 25의 (A) 및 도 25의 (B)는 본 발명의 정보 처리 시스템에 있어서의 유효화 키 블록(EKB)의 생성 과정의 설명도.

도 26의 (A) 및 도 26의 (B)는 본 발명의 정보 처리 시스템에 있어서의 간략화 유효화 키 블록(EKB)(예1)의 설명도.

도 27의 (A) 내지 도 27의 (C)는 본 발명의 정보 처리 시스템에 있어서의 계층 트리 구조의 엔티티 관리 구성에 대한 설명도.

도 28의 (A) 내지 도 28의 (C)는 본 발명의 정보 처리 시스템에 있어서의 계층 트리 구조의 엔티티 관리 구성의 상세에 대한 설명도.

도 29A 및 도 29B는 본 발명의 정보 처리 시스템에 있어서의 계층 트리 구조의 엔티티 관리 구성에 대한 설명도.

도 30은 본 발명의 정보 처리 시스템에 있어서의 계층 트리 구조의 엔티티 관리 구성에서의 리저브 노드에 대한 설명도.

도 31은 본 발명의 정보 처리 시스템에 있어서의 계층 트리 구조의 엔티티 관리 구성에서의 신규 엔티티 등록 처리 시퀀스에 대한 설명도.

도 32는 본 발명의 정보 처리 시스템에 있어서의 계층 트리 구조의 엔티티 관리 구성에서의 신규 엔티티와 상위 엔티티의 관계에 대한 설명도.

도 33A 및 도 33B는 본 발명의 정보 처리 시스템에 있어서의 계층 트리 구조의 엔티티 관리 구성으로 이용하는 서브 EKB에 대한 설명도.

도 34의 (A) 내지 도 34의 (D)는 본 발명의 정보 처리 시스템에 있어서의 계층 트리 구조의 엔티티 관리 구성에서의 디바이스 리보크 처리에 대한 설명도.

도 35는 본 발명의 정보 처리 시스템에 있어서의 계층 트리 구조의 엔티티 관리 구성에서의 디바이스 리보크 처리 시퀀스에 대한 설명도.

도 36A 및 도 36B는 본 발명의 정보 처리 시스템에 있어서의 계층 트리 구조의 엔티티 관리 구성에서의 디바이스 리보크 시의 갱신 서브 EKB에 대한 설명도.

도 37의 (A) 내지 도 37의 (D)는 본 발명의 정보 처리 시스템에 있어서의 계층 트리 구조의 엔티티 관리 구성에서의 엔티티 리보크 처리에 대한 설명도.

도 38은 본 발명의 정보 처리 시스템에 있어서의 계층 트리 구조의 엔티티 관리 구성에서의 엔티티 리보크 처리 시퀀스에 대한 설명도.

도 39는 본 발명의 정보 처리 시스템에 있어서의 계층 트리 구조의 엔티티 관리 구성에서의 리보크 엔티티와 상위 엔티티의 관계에 대한 설명도.

도 40은 본 발명의 정보 처리 시스템에 있어서의 계층 트리 구조의 엔티티 관리 구성에서의 캐이퍼빌리티 설정에 대한 설명도.

도 41은 본 발명의 정보 처리 시스템에 있어서의 계층 트리 구조의 엔티티 관리 구성에서의 캐이퍼빌리티 설정에 대한 설명도.

도 42의 (A) 및 도 42의 (B)는 본 발명의 정보 처리 시스템에 있어서의 키 발행 센터(KDC)가 관리하는 캐이퍼빌리티 관리 테이블 구성의 설명도.

도 43은 본 발명의 정보 처리 시스템에 있어서의 키 발행 센터(KDC)가 관리하는 캐이퍼빌리티 관리 테이블에 기초한 EKB 생성 처리 흐름도.

도 44는 본 발명의 정보 처리 시스템에 있어서의 신규 엔티티 등록 시의 캐이퍼빌리티 통지 처리의 설명도.

〈발명을 실시하기 위한 최량의 형태〉

[시스템 개요]

도 1에 본 발명의 데이터 처리 시스템이 적용 가능한 콘텐츠 배신 시스템예를 나타낸다. 콘텐츠 배신측(10)은 콘텐츠 수신측(20)이 갖는 여러가지 콘텐츠 재생 가능 기기에 대하여 콘텐츠, 또는 콘텐츠 키를 암호화하여 송신한다. 수신측 (20)에서의 기기에서는 수신한 암호화 콘텐츠, 또는 암호화 콘텐츠 키 등을 복호하여 콘텐츠 또는 콘텐츠 키를 취득하여, 화상 데이터, 음성 데이터의 재생, 또는 각종 프로그램의 실행 등을 행한다. 콘텐츠 배신측(10)과 콘텐츠 수신측(20) 간의 데이터 교환은 인터넷 등의 네트워크를 통해, 또는 DVD, CD 등의 유통 가능한 기억 매체를 통해 실행된다.

콘텐츠 배신측(10)의 데이터 배신 수단으로서는 인터넷(11), 위성 방송(12), 전화 회선(13), DVD, CD 등의 미디어(14) 등이 있으며, 한편, 콘텐츠 수신측(20)의 디바이스로서는 퍼스널 컴퓨터(PC: 21), 포터블 디바이스(PD: 22), 휴대 전화, PDA (Personal Digital Assistants) 등의 휴대 기기(23), DVD, CD 플레이어 등의 기록 재생기(24), 게임 단말기 등의 재생 전용기(25) 등이 있다. 이들 콘텐츠 수신측(20)의 각 디바이스는 콘텐츠 배신측(10)으로부터 제공된 콘텐츠를 네트워크 등의 통신 수단 또는 미디어(30)로부터 취득한다.

[디바이스 구성]

도 2에, 도 1에 도시한 콘텐츠 수신측(20)의 디바이스의 일례로서, 기록 재생 장치(100)의 구성 블록도를 도시한다. 기록 재생 장치(100)는 입출력 I/F (Interface: 120), MPEG(Moving Picture Experts Group) 코덱(130), A/D, D/A 컨버터(141)를 구비한 입출력 I/F(Interface: 140), 암호 처리 수단(150), ROM(Read Only Memory: 160), CPU(Central Processing Unit: 170), 메모리(180), 기록 매체(195)의 드라이브(190)를 갖고, 이들은 버스(110)에 의해 서로 접속되어 있다.

입출력 I/F(120)는 외부로부터 공급되는 화상, 음성, 프로그램 등의 각종 콘텐츠를 구성하는 디지털 신호를 수신하고, 버스(110) 상으로 출력함과 함께,

버스(110) 상의 디지털 신호를 수신하여 외부로 출력한다. MPEG 코덱(130)은 버스(110)를 통해 공급되는 MPEG 부호화된 데이터를 MPEG 디코드하고, 입출력 I/F(140)로 출력함과 함께, 입출력 I/F(140)로부터 공급되는 디지털 신호를 MPEG 인코드하여 버스(110) 상으로 출력한다. 입출력 I/F(140)는 A/D, D/A 컨버터(141)를 내장하고 있다. 입출력 I/F(140)는 외부로부터 공급되는 콘텐츠로서의 아날로그 신호를 수신하고, A/D, D/A 컨버터(141)로 A/D(Analog Digital) 변환함으로써, 디지털 신호로서, MPEG 코덱(130)으로 출력함과 함께, MPEG 코덱(130)으로부터의 디지털 신호를 A/D, D/A 컨버터(141)로 D/A(Digital Analog) 변환함으로써, 아날로그 신호로서, 외부로 출력한다.

암호 처리 수단(150)은 예를 들면, 1칩의 LSI(Large Scale Integrated Circuit)로 구성되고, 버스(110)를 통해 공급되는 콘텐츠로서의 디지털 신호의 암호화, 복호 처리, 또는 인증 처리를 실행하고, 암호 데이터, 복호 데이터 등을 버스(110) 상으로 출력하는 구성을 갖는다. 또, 암호 처리 수단(150)은 1칩 LSI에 한하지 않고, 각종 소프트웨어 또는 하드웨어를 조합한 구성에 의해 실현할 수도 있다. 소프트웨어 구성에 의한 처리 수단으로서의 구성에 대해서는 후단에서 설명한다.

ROM(160)은 기록 재생 장치에 의해 처리되는 프로그램 데이터를 저장한다. CPU(170)는 ROM(160), 메모리(180)에 기억된 프로그램을 실행함으로써, MPEG 코덱 (130)이나 암호 처리 수단(150) 등을 제어한다. 메모리(180)는 예를 들면, 불휘발성 메모리로, CPU(170)가 실행하는 프로그램이나, CPU(170)의 동작 상 필요한 데이터, 또한 디바이스에 의해 실행되는 암호 처리에 사용되는 키 세트를 기억한다. 키 세트에 대해서는 후단에서 설명한다. 드라이브(190)는 디지털 데이터를 기록 재생할 수 있는 기록 매체(195)를 구동함으로써, 기록 매체(195)로부터 디지털 데이터를 판독하여(재생하여), 버스(110) 상으로 출력함과 함께, 버스(110)를 통해 공급되는 디지털 데이터를 기록 매체(195)에 공급하여 기록시킨다.

기록 매체(195)는 예를 들면, DVD, CD 등의 광 디스크, 광 자기 디스크, 자기 디스크, 자기 테이프, 또는 RAM 등의 반도체 메모리 등의 디지털 데이터 기억 가능 매체로서, 본 실시의 형태에서는 드라이브(190)에 대하여 착탈 가능한 구성으로 한다. 단, 기록 매체(195)는 기록 재생 장치(100)에 내장하는 구성으로 해도 무방하다.

또, 도 2에 도시한 암호 처리 수단(150)은 하나의 원 칩 LSI로서 구성해도 되고, 또한 소프트웨어, 하드웨어를 조합한 구성에 의해 실현하는 구성으로 해도 무방하다.

[키 배신 구성으로서의 트리(나무) 구조에 대하여]

다음으로, 도 1에 도시한 콘텐츠 배신측(10)으로부터 콘텐츠 수신측(20)의 각 디바이스에 암호 데이터를 배신하는 경우에 있어서의 각 디바이스에 있어서의 암호 처리 키의 보유 구성 및 데이터 배신 구성을 도 3을 이용하여 설명한다.

도 3의 최하단에 도시한 번호 0~15가 콘텐츠 수신측(20)의 개개의 디바이스이다. 즉, 도 3에 도시한 계층 트리(나무) 구조의 각 잎(리프: leaf)이 각각의 디바이스에 상당한다.

각 디바이스 0~15는 제조 시 또는 출하 시, 또는 그 후에 있어서, 도 3에 도시한 계층 트리(나무) 구조에 있어서의, 자신의 리프로부터 루트에 이르기까지의 노드에 할당된 키(노드 키) 및 각 리프의 리프 키를 포함하는 키 세트를 메모리에 저장한다. 도 3의 최하단에 도시한 K0000~K1111이 각 디바이스 0~15에 각각 할당된 리프 키이고, 최상단의 KR(루트 키)로부터, 최하단으로부터 2번째 절(노드)에 기재된 키: KR~K111을 노드 키로 한다.

도 3에 도시한 트리 구성에서, 예를 들면 디바이스 0은 리프 키 K0000과, 노드 키: K000, K00, K0, KR을 소유한다. 디바이스 5는 K0101, K010, K01, K0, KR을 소유한다. 디바이스 15는 K1111, K111, K11, K1, KR을 소유한다. 또, 도 3의 트리에는 디바이스가 0~15의 16개만 기재되고, 트리 구조도 4단 구성의 균형 잡힌 좌우 대칭 구성으로서 나타내고 있지만, 더 많은 디바이스가 트리 중에 구성되어, 트리의 각 부에서 다른 단수 구성을 더 가질 수 있다.

또한, 도 3의 트리 구조에 포함되는 각 디바이스에는 여러가지 기록 매체, 예를 들면, 디바이스 매립형 또는 디바이스에 착탈 가능하게 구성된 DVD, CD, MD, 플래시 메모리 등을 사용하는 여러가지 타입의 디바이스가 포함되어 있다. 또한, 여러가지 어플리케이션 서비스가 공존 가능하다. 이러한 다른 디바이스, 다른 어플리케이션의 공존 구성 상에 도 3에 도시한 콘텐츠 또는 키 배포 구성인 계층 트리 구조가 적용된다.

이들 여러가지 디바이스, 어플리케이션이 공존하는 시스템에 있어서, 예를 들면 도 3의 점선으로 둘러싼 부분, 즉 디바이스 0, 1, 2, 3을 동일한 기록 매체를 이용하는 하나의 그룹으로서 설정한다. 예를 들면, 이 점선으로 둘러싼 그룹 내에 포함되는 디바이스에 대해서는, 통합하여 공통의 콘텐츠를 암호화하여 프로바이더로부터 송부하거나, 각 디바이스에 공통으로 사용하는 콘텐츠 키를 송부하거나, 또는 각 디바이스로부터 프로바이더 또는 결제 기관 등에 콘텐츠 요금의 지불 데이터를 역시 암호화하여 출력하는 처리가 실행된다. 콘텐츠 프로바이더, 또는 결제 처리 기관 등, 각 디바이스와의 데이터 송수신을 행하는 기관은 도 3의 점선으로 둘러싼 부분, 즉 디바이스 0, 1, 2, 3을 하나의 그룹으로서 일괄적으로 데이터를 송부하는 처리를 실행한다. 이러한 그룹은 도 3의 트리 중에 복수 존재한다. 콘텐츠 프로바이더, 또는 결제 처리 기관 등, 각 디바이스와의 데이터 송수신을 행하는 기관은 메시지 데이터 배신 수단으로서 기능한다.

또, 노드 키, 리프 키는 임의의 하나의 키 관리 센터에 의해 통괄적으로 관리해도 되고, 각 그룹에 대한 여러가지 데이터 송수신을 행하는 프로바이더, 결제 기관 등의 메시지 데이터 배신 수단에 의해 그룹마다 관리하는 구성으로 해도 무방하다. 이들 노드 키, 리프 키는, 예를 들면 키 누설 등의 경우에 갱신 처리가 실행되고, 이 갱신 처리는 키 관리 센터, 프로바이더, 결제 기관 등이 실행한다.

이 트리 구조에 있어서, 도 3에서 분명한 바와 같이 하나의 그룹에 포함되는 3개의 디바이스 0, 1, 2, 3은 노드 키로서 공통의 키 K00, K0, KR을 보유한다. 이 노드 키 공유 구성을 이용함으로써, 예를 들면 공통의 콘텐츠 키를 디바이스 0, 1, 2, 3에만 제공할 수 있다. 예를 들면, 공통으로 보유하는 노드 키 K00 자체를 콘텐츠 키로서 설정하면, 새로운 키 송부를 실행하지 않고 디바이스 0, 1, 2, 3만이 공통의 콘텐츠 키의 설정이 가능하다. 또한, 새로운 콘텐츠 키 Kcon을 노드 키 K00으로 암호화한 값 Enc(K00, Kcon)를 네트워크를 통해, 또는 기록 매체에 저장하여 디바이스 0, 1, 2, 3에 배포하면, 디바이스 0, 1, 2, 3만이 각각의 디바이스에 있어서 보유하는 공유 노드 키 K00을 이용하여 암호 Enc(K00, Kcon)를 풀어 콘텐츠 키: Kcon을 얻을 수 있다. 또, Enc(Ka, Kb)는 Kb를 Ka에 의해 암호화한 데이터인 것을 나타낸다.

또한, 임의 시점 t에서, 디바이스 3이 소유하는 키: K0011, K001, K00, K0, KR이 공격자(해커)에 의해 해석되어 노출된 것이 발각한 경우, 그 이후, 시스템(디바이스 0, 1, 2, 3의 그룹)으로 송수신되는 데이터를 지키기 위해서, 디바이스 3을 시스템으로부터 분리할 필요가 있다. 그 때문에는 노드 키: K001, K00, K0, KR을 각각 새로운 키 K(t)001, K(t)00, K(t)0, K(t)R로 갱신하고, 디바이스 0, 1, 2에 그 갱신 키를 전할 필요가 있다. 여기서, K(t)aaa는 키 Kaaa의 세대(Generation) : t의 갱신 키인 것을 나타낸다.

갱신 키의 배포 처리에 대하여 설명한다. 키의 갱신은 예를 들면, 도 4A에 도시한 유효화 키 블록(EKB: Enabling Key Block)이라 불리는 블록 데이터에 의해 구성되는 데이블을, 예를 들면 네트워크, 또는 기록 매체에 저장하여 디바이스 0, 1, 2에 공급함으로써 실행된다. 또, 유효화 키 블록(EKB)은 도 3에 도시한 바와 같은 트리 구조를 구성하는 각 리프에 대응하는 디바이스에 새롭게 갱신된 키를 배포하기 위한 암호화 키에 의해 구성된다. 유효화 키 블록(EKB)은 키 갱신 블록(KRB: Key Renewal Block)이라 불리는 경우도 있다.

도 4A에 도시한 유효화 키 블록(EKB)에는 노드 키의 갱신이 필요한 디바이스만이 갱신 가능한 데이터 구성을 갖는 블록 데이터로서 구성된다. 도 4A 및 도 4B의 예는 도 3에 도시한 트리 구조 중의 디바이스 0, 1, 2에 있어서, 세대 t의 갱신 노드 키를 배포하는 것을 목적으로 하여 형성된 블록 데이터이다. 도 3에서 분명한 바와 같이 디바이스 0, 디바이스 1은 갱신 노드 키로서 K(t)00, K(t)0, K(t)R이 필요하고, 디바이스 2는 갱신 노드 키로서 K(t)001, K(t)00, K(t)0, K(t)R이 필요하다.

도 4A의 EKB에 도시된 바와 같이 EKB에는 복수의 암호화 키가 포함된다. 최하단의 암호화 키는 Enc(K0010, K(t)001)이다. 이는 디바이스 2가 갖는 리프 키 K0010에 의해 암호화된 갱신 노드 키 K(t)001이고, 디바이스 2는 자신이 갖는 리프 키에 의해 이 암호화 키를 복호하고, K(t)001을 얻을 수 있다. 또한, 복호에 의해 얻은 K(t)001을 이용하여, 도 4A의 아래로부터 2단째 암호화 키 Enc(K(t)001, K(t)00)를 복호할 수 있게 되어, 갱신 노드 키 K(t)00을 얻을 수 있다. 이하, 순차적으로 도 4A의 위로부터 2단째 암호화 키 Enc(K(t)00, K(t)0)를 복호하고, 갱신 노드 키 K(t)0, 도 4A의 위로부터 1단째 암호화 키 Enc(K(t)0, K(t)R)를 복호하여 K(t)R을 얻는다. 한편, 디바이스 K0000. K0001은, 노드 키 K000은 갱신 대상에 포함되지 않고, 갱신 노드 키로서 필요한 것은 K(t)00, K(t)0, K(t)R이다. 디바이스 K0000. K0001은 도 4A의 위로부터 3단째 암호화 키 Enc(K000, K(t)00)를 복호하여, K(t)00을 취득하고, 이하, 도 4A의 위로부터 2단째 암호화 키 Enc (K(t)00, K(t)0)를 복호하고, 갱신 노드 키 K(t)0, 도 4A의 위로부터 1단째 암호화 키 Enc(K(t)0, K(t)R)를 복호하고 K(t)R을 얻는다. 이와 같이 하여, 디바이스 0, 1, 2는 갱신한 키 K(t)001, K(t)00, K(t)0, K(t)R을 얻을 수 있다. 또, 도 4A의 인덱스는 복호 키로서 사용하는 노드 키, 리프 키의 절대 번지를 나타낸다.

도 3에 도시한 트리 구조의 상위 단의 노드 키: K(t)0, K(t)R의 갱신이 불필요하고, 노드 키 K00만의 갱신 처리가 필요한 경우에는 도 4B의 유효화 키 블록(EKB)을 이용함으로써, 갱신 노드 키 K(t)00을 디바이스 0, 1, 2에 배포할 수 있다.

도 4B에 도시한 EKB는, 예를 들면 특정한 그룹에서 공유하는 새로운 콘텐츠 키를 배포하는 경우에 이용 가능하다. 구체예로서, 도 3에 점선으로 도시한 그룹 내의 디바이스 0, 1, 2, 3이 임의의 기록 매체를 이용하고 있으며, 새로운 공통의 콘텐츠 키 K(t)con이 필요하다고 한다. 이 때, 디바이스 0, 1, 2, 3의 공통의 노드 키 K00을 갱신한 K(t)00을 이용하여 새로운 공통의 갱신 콘텐츠 키: K(t)con을 암호화한 데이터 Enc(K(t), K(t)con)를 도 4B에 도시한 EKB와 함께 배포한다. 이 배포에 의해, 디바이스 4 등, 그 밖의 그룹의 기기에 있어서는 복호되지 않는 데이터로서의 배포가 가능하게 된다.

즉, 디바이스 0, 1, 2는 EKB를 처리하여 얻은 K(t)00을 이용하여 상기 암호문을 복호하면, t 시점에서의 콘텐츠 키 K(t)con을 얻을 수 있다.

[EKB를 사용한 콘텐츠 키의 배포]

도 5에, t 시점에서의 콘텐츠 키 K(t)con을 얻는 처리예로서, K(t)00을 이용하여 새로운 공통의 콘텐츠 키 K(t)con을 암호화한 데이터 Enc(K(t)00, K(t)con)와 도 4B에 도시한 EKB를 기록 매체를 통해 수령한 디바이스 0의 처리를 나타낸다. 즉, EKB에 의한 암호화 메시지 데이터를 콘텐츠 키 K(t)con으로 한 예이다.

도 5에 도시한 바와 같이 디바이스 0은 기록 매체에 저장되어 있는 세대: t 시점의 EKB와 자신이 사전에 저장하고 있는 노드 키 K000을 이용하여 상술한 바와 마찬가지의 EKB 처리에 의해, 노드 키 K(t)00을 생성한다. 또한, 복호한 갱신 노드 키 K(t)00을 이용하여 갱신 콘텐츠 키 K(t)con을 복호하고, 후에 그것을 사용하기 위해서 자신만이 갖는 리프 키 K0000으로 암호화하여 저장한다.

[EKB의 포맷]

도 6에 유효화 키 블록(EKB)의 포맷예를 나타낸다. 버전(601)은 유효화 키 블록(EKB)의 버전을 나타내는 식별자이다. 또, 버전은 최신의 EKB를 식별하는 기능과 콘텐츠와의 대응 관계를 나타내는 기능을 갖는다. 깊이(depth)는 유효화 키 블록(EKB)의 배포처의 디바이스에 대한 계층 트리의 계층 수를 나타낸다. 데이터 포인터(603)는 유효화 키 블록(EKB) 중의 데이터부의 위치를 나타내는 포인터이고, 태그 포인터(604)는 태그부의 위치, 서명 포인터(605)는 서명의 위치를 나타내는 포인터이다.

데이터부(606)는, 예를 들면 갱신하는 노드 키를 암호화한 데이터를 저장한다. 예를 들면, 도 5에 도시한 바와 같은 갱신된 노드 키에 관한 각 암호화 키 등을 저장한다.

태그부(607)는 데이터부에 저장된 암호화된 노드 키, 리프 키의 위치 관계를 나타내는 태그이다. 이 태그의 부여 룰을 도 7의 (A) 내지 도 7의 (B)를 이용하여 설명한다. 도 7의 (A) 내지 도 7의 (B)에서는 데이터로서 먼저 도 4A에서 설명한 유효화 키 블록(EKB)을 송부하는 예를 나타내고 있다. 이 때의 데이터는, 도 7의 (B)에 도시한 바와 같다. 이 때의 암호화 키에 포함되는 톱 노드의 어드레스를 톱 노드 어드레스로 한다. 이 경우에는 루트 키의 갱신 키 K(t)R이 포함되어 있기 때문에, 톱 노드 어드레스는 KR이 된다. 이 때, 예를 들면 최상단의 데이터 Enc(K(t)0, K(t)R)는 도 7의 (A)에 도시한 계층 트리에 나타내는 위치에 있다. 여기서, 다음의 데이터는 Enc(K(t)00, K(t)0)로서, 트리 상에서는 이전의 데이터의 좌측 아래의 위치에 있다. 데이터가 있는 경우에는 태그가 0, 없는 경우에는 1이 설정된다. 태크는 {좌측(L) 태그, 우측(R) 태그}로서 설정된다. 최상단의 데이터 Enc(K(t)0, K(t)R)의 좌측에는 데이터가 있기 때문에, L 태그=0, 우측에는 데이터가 없기 때문에, R 태그=1이 된다. 이하, 모든 데이터에 태그가 설정되고, 도 7 (C)에 도시한 데이터 열, 및 태그 열이 구성된다.

태그는 데이터 Enc(Kxxx, Kyyy)가 트리 구조의 어디에 위치하고 있는 것인지를 나타내기 위해서 설정되는 것이다. 데이터부에 저장되는 키 데이터 Enc(Kxxx, Kyyy) ... 는, 단순하게 암호화된 키의 나열 데이터에 불과하므로, 상술한 태그에 의해 데이터로서 저장된 암호화 키의 트리 상의 위치를 판별 가능하게 한 것이다. 상술한 태그를 이용하지 않고, 앞의 도 4A 및 도 4B에서 설명한 구성과 같이 암호화 데이터에 대응시킨 노드·인덱스를 이용하여, 예를 들면,

0:Enc(K(t)0, K(t)root)

00:Enc(K(t)00, K(t)0)

000:Enc(K(t)000, K(t)00)

... 와 같은 데이터 구성으로 할 수도 있지만, 이러한 인덱스를 이용한 구성으로 하면 리던던시 데이터가 되어 데이터량이 증대하여, 네트워크를 통한 배신 등에 있어서는 바람직하지 않다. 이에 대하여, 상술한 태그를 키 위치를 나타내는 색인 데이터로 이용함으로써, 적은 데이터량으로 키 위치의 판별이 가능하게 된다.

도 6으로 되돌아가 EKB 포맷에 대하여 다시 설명한다. 서명(Signature)은 유효화 키 블록(EKB)을 발행한, 예를 들면 키 관리 센터, 콘텐츠 프로바이더, 결제 기관 등이 실행하는 전자 서명이다. EKB를 수령한 디바이스는 서명 검증에 의해 정당한 유효화 키 블록(EKB) 발행자가 발행한 유효화 키 블록(EKB)인 것을 확인한다.

[EKB를 사용한 콘텐츠 키 및 콘텐츠의 배신]

상술한 예에서는 콘텐츠 키만을 EKB와 함께 송부하는 예에 대하여 설명했지만, 콘텐츠 키로 암호화한 콘텐츠와, 콘텐츠 키 암호 키로 암호화한 콘텐츠 키와, EKB에 의해 암호화한 콘텐츠 키 암호 키를 함께 송부하는 구성에 대하여 이하 설명한다.

도 8A 및 도 8B에 이 데이터 구성을 나타낸다. 도 8A에 도시한 구성에서, Enc(Kcon, content: 801)는 콘텐츠(Content)를 콘텐츠 키(Kcon)로 암호화한 데이터이고, Enc(KEK, Kcon: 802)는 콘텐츠 키(Kcon)를 콘텐츠 키 암호 키(KEK: Key Encryption Key)로 암호화한 데이터이고, Enc(EKB, KEK: 803)는 콘텐츠 키 암호 키 KEK를 유효화 키 블록(EKB)에 의해 암호화한 데이터인 것을 나타낸다.

여기서, 콘텐츠 키 암호 키 KEK는 도 3에서 도시한 노드 키(K000, K00 …), 또는 루트 키(KR) 자체이어도 되고, 또한 노드 키(K000, K00 …), 또는 루트 키(KR)에 의해 암호화된 키이어도 된다.

도 8B는 복수의 콘텐츠가 미디어에 기록되고, 각각이 동일한 Enc(EKB, KEK: 805)를 이용하고 있는 경우의 구성예를 나타내는데, 이러한 구성에서는 각 데이터에 동일한 Enc(EKB, KEK)를 부가하지 않고, Enc(EKB, KEK)에 링크하는 링크처를 나타내는 데이터를 각 데이터에 부가하는 구성으로 할 수 있다.

도 9에 콘텐츠 키 암호 키 KEK를 도 3에 도시한 노드 키 K00을 갱신한 갱신 노드 키 K(t)00으로서 구성한 경우의 예를 나타낸다. 이 경우, 도 3의 점선 프레임으로 둘러싼 그룹에서 디바이스 3이, 예를 들면 키 누설에 의해 리보크(배제)되어 있다고 해서, 다른 그룹의 멤버, 즉, 디바이스 0, 1, 2에 대하여 도 9에 도시한 유효화 키 블록(EKB)과, 콘텐츠 키(Kcon)를 콘텐츠 키 암호 키(KEK=K(t)00)로 암호화한 데이터와, 콘텐츠(content)를 콘텐츠 키(Kcon)로 암호화한 데이터를 배신함으로써, 디바이스 0, 1, 2는 콘텐츠를 얻을 수 있다.

도 9의 우측에는 디바이스 0에서의 복호 순서를 나타내고 있다. 디바이스 0은 우선, 수령한 유효화 키 블록으로부터 자신이 보유한 리프 키 K000을 이용한 복호 처리에 의해, 콘텐츠 키 암호 키(KEK=K(t)00)를 취득한다. 다음으로, K(t)00에 의한 복호에 의해 콘텐츠 키 Kcon을 취득하고, 또한 콘텐츠 키 Kcon에 의해 콘텐츠 복호를 행한다. 이들 처리에 의해, 디바이스 0은 콘텐츠를 이용할 수 있게 된다. 디바이스 1, 2에 있어서도 각각 다른 처리 순서로 EKB를 처리함으로써, 콘텐츠 키 암호 키(KEK=K(t)00)를 취득할 수 있고, 마찬가지로 콘텐츠를 이용할 수 있다.

도 3에 도시한 다른 그룹의 디바이스 4, 5, 6 …는, 이와 유사한 데이터 (EKB)를 수신했다고 해도, 자신이 보유한 리프 키, 노드 키를 이용하여 콘텐츠 키 암호 키(KEK=K(t)00)를 취득할 수 없다. 마찬가지로 리보크된 디바이스 3에 있어서도, 자신이 보유한 리프 키, 노드 키에서는 콘텐츠 키 암호 키(KEK=K(t)00)를 취득할 수 없고, 정당한 권리를 갖는 디바이스만이 콘텐츠를 복호하여 이용할 수 있다.

이와 같이 EKB를 이용한 콘텐츠 키의 배송을 이용하면, 데이터량을 적게 하고, 또한 안전하게 정당 권리자만이 복호 가능하게 한 암호화 콘텐츠를 배신할 수 있다.

또, 유효화 키 블록(EKB), 콘텐츠 키, 암호화 콘텐츠 등은 네트워크를 통해 안전하게 배신할 수 있는 구성이지만, 유효화 키 블록(EKB), 콘텐츠 키, 암호화 콘텐츠를 DVD, CD 등의 기록 매체에 저장하여 사용자에게 제공할 수도 있다. 이 경우, 기록 매체에 저장된 암호화 콘텐츠의 복호에는 동일한 기록 매체에 저장된 유효화 키 블록(EKB)의 복호에 의해 얻어지는 콘텐츠 키를 사용하도록 구성하면, 사전에 정당 권리자만이 보유하는 리프 키, 노드 키에 의해서만 이용 가능한 암호화 콘텐츠의 배포 처리, 즉 이용 가능한 사용자 디바이스를 한정한 콘텐츠 배포가 간이한 구성으로 실현 가능하다.

도 10에 기록 매체에 암호화 콘텐츠와 함께 유효화 키 블록(EKB)을 저장한 구성예를 나타낸다. 도 10에 도시한 예에서는 기록 매체에 콘텐츠 C1～C4가 저장되고, 또한 각 저장 콘텐츠에 대응하는 유효화 키 블록(EKB)을 대응시킨 데이터가 저장되고, 또한 버전 M의 유효화 키 블록(EKB_M)이 저장되어 있다. 예를 들면, EKB_1은 콘텐츠 C1을 암호화한 콘텐츠 키 Kcon1을 생성하는 데 사용되고, 예를 들면, EKB_2는 콘텐츠 C2를 암호화한 콘텐츠 키 Kcon2를 생성하는 데 사용된다. 본 예에서는 버전 M의 유효화 키 블록(EKB_M)이 기록 매체에 저장되어 있고, 콘텐츠 C3, C4는 유효화 키 블록(EKB-M)에 대응되어 있기 때문에, 유효화 키 블록(EKB_M)의 복호에 의해 콘텐츠 C3, C4의 콘텐츠 키를 취득할 수 있다. EKB_1, EKB_2는 디스크에 저장되어 있지 않기 때문에, 새로운 제공 수단, 예를 들면 네트워크 배신, 또는 기록 매체에 의한 배신에 의해 각각의 콘텐츠 키를 복호하기 위해서 필요한 EKB_1, EKB_2를 취득할 필요가 있다.

도 11A 및 도 11B에, 복수의 디바이스 사이에서 콘텐츠 키가 유통하는 경우의 EKB를 이용한 콘텐츠 키의 배신과, 종래의 콘텐츠 키 배신 처리의 비교예를 나타낸다. 도 11A는 종래 구성이고, 도 11B는 본 발명의 유효화 키 블록(EKB)을 이용한 예이다. 또, 도 11A 및 도 11B에서 Ka(Kb)는 Kb를 Ka로 암호화한 데이터인 것을 나타낸다.

도 11A에 도시한 바와 같이 종래는 데이터 송수신자의 정당성을 확인하고, 또한 데이터 송신의 암호화 처리에 사용하는 세션 키 Kses를 공유하기 위해서 각 디바이스 사이에서, 인증 처리 및 키 교환 처리(AKE: Authentication and Key Exchange)를 실행하고, 인증이 성립한 것을 조건으로 하여 세션 키 Kses로 콘텐츠 키 Kcon을 암호화하여 송신하는 처리를 행하고 있었다.

예를 들면, 도 11A의 PC에서는 수신한 세션 키로 암호화한 콘텐츠 키 Kses(Kcon)를 세션 키로 복호하여 Kcon을 얻을 수 있고, 또한 취득한 Kcon을 PC 자체가 보유하는 보존 키 Kstr로 암호화하여 자신의 메모리에 보존할 수 있다.

도 11A에서, 콘텐츠 프로바이더는 도 11A의 기록 디바이스(1101)에만 데이터를 이용 가능한 형태로 배신하고자 하는 경우라도, PC, 재생 장치가 사이에 존재하는 경우에는 도 11A에 도시한 바와 같이 인증 처리를 실행하고, 각각의 세션 키로 콘텐츠 키를 암호화하여 배신하는 처리가 필요하게 된다. 또한, 사이에 개재하는 PC, 재생 장치에서도 인증 처리에 있어서 생성하여 공유하게 된 세션 키를 이용함으로써 암호화 콘텐츠 키를 복호하여 콘텐츠 키를 취득할 수 있다.

한편, 도 11B의 하단에 도시한 유효화 키 블록(EKB)을 이용한 예에 있어서는 콘텐츠 프로바이더로부터 유효화 키 블록(EKB)과, 유효화 키 블록(EKB)의 처리에 의해 얻어지는 노드 키, 또는 루트 키에 의해 콘텐츠 키 Kcon을 암호화한 데이터(도면의 예에서는 Kroot(Kcon))를 배신함으로써, 배신한 EKB의 처리가 가능한 기기에 있어서만 콘텐츠 키 Kcon을 복호하여 취득할 수 있다.

따라서, 예를 들면 도 11B의 오른쪽 단에만 이용 가능한 유효화 키 블록 (EKB)을 생성하여, 그 유효화 키 블록(EKB)과, 그 EKB 처리에 의해 얻어지는 노드 키, 또는 루트 키에 의해 콘텐츠 키 Kcon을 암호화한 데이터를 함께 송신함으로써, 사이에 존재하는 PC, 재생 기기 등은 자신이 갖는 리프 키, 노드 키에 의해서는 EKB의 처리를 실행할 수 없다. 따라서, 데이터 송수신 디바이스 사이에서의 인증 처리, 세션 키의 생성, 세션 키에 의한 콘텐츠 키 Kcon의 암호화 처리 등의 처리를 실행하지 않고, 안전하게 정당한 디바이스에 대해서만 이용 가능한 콘텐츠 키를 배신할 수 있다.

PC, 기록 재생기에도 이용 가능한 콘텐츠 키를 배신하고자 하는 경우에는 각각에 있어서 처리 가능한 유효화 키 블록(EKB)을 생성하여 배신함으로써, 공통의 콘텐츠 키를 취득할 수 있다.

[유효화 키 블록(EKB)을 사용한 인증 키의 배신(공통 키 방식)]

상술한 유효화 키 블록(EKB)을 사용한 데이터 또는 키의 배신에 있어서, 디바이스 사이에서 전송되는 유효화 키 블록(EKB) 및 콘텐츠 또는 콘텐츠 키는 항상 동일한 암호화 형태를 유지하고 있기 때문에, 데이터 전송로를 이용하여 기록하고, 재차 후에 전송하는, 소위 리플레이 어택(replay attack)에 의해 부정 복사가 생성될 가능성이 있다. 이를 방지하는 구성으로서는 데이터 전송 디바이스 사이에서, 종래와 마찬가지의 인증 처리 및 키 교환 처리를 실행하는 것이 유효한 수단이다. 여기서는 이 인증 처리 및 키 교환 처리를 실행할 때에 사용하는 인증 키 Kake를 상술한 유효화 키 블록(EKB)을 사용하여 디바이스에 배신함으로써, 안전한 비밀 키로서 공유하는 인증 키를 갖고, 공통 키 방식에 따른 인증 처리를 실행하는 구성에 대하여 설명한다. 즉, EKB에 의한 암호화 메시지 데이터를 인증 키로 한 예이다.

도 12에, 공통 키 암호 방식을 이용한 상호 인증 방법(ISO/IEC 9798-2)을 나타낸다. 도 12에서는 공통 키 암호 방식으로서 DES를 이용하고 있지만, 공통 키 암호 방식이면 다른 방식도 가능하다. 도 12에서, 우선, B가 64 비트의 난수 Rb를 생성하고, Rb 및 자신의 ID인 ID(b)를 A에 송신한다. 이를 수신한 A는 새롭게 64비트의 난수 Ra를 생성하고, Ra, Rb, ID(b) 순으로, DES의 CBC 모드로 키 Kab를 이용하여 데이터를 암호화하여, B에 반송한다. 또, 키 Kab는 A 및 B에 공통의 비밀 키로서 각각의 기록 소자 내에 저장하는 키이다. DES의 CBC 모드를 이용한 키 Kab에 의한 암호화 처리는, 예를 들면 DES를 이용한 처리에 있어서는 초기치와 Ra와의 배타적 논리합을 구하고, DES 암호화부에서 키 Kab를 이용하여 암호화하여 암호문 E1을 생성하고, 계속해서 암호문 E1과 Rb와의 배타적 논리합을 구하고, DES 암호화부에서 키 Kab를 이용하여 암호화하여 암호문 E2를 생성하고, 또한 암호문 E2와 ID(b)와의 배타적 논리합을 구하고, DES 암호화부에서 키 Kab를 이용하여 암호화하여 생성한 암호문 E3에 따라 송신 데이터(Token-AB)를 생성한다.

이를 수신한 B는 수신 데이터를 역시 공통의 비밀 키로서 각각의 기록 소자 내에 저장하는 키 Kab(인증 키)로 복호화한다. 수신 데이터의 복호화 방법은 우선, 암호문 E1을 인증 키 Kab로서 복호화하고, 난수 Ra를 얻는다. 다음으로, 암호문 E2를 인증 키 Kab로 복호화하고, 그 결과와 E1과의 배타적 논리합을 구하여 Rb를 얻는다. 마지막으로, 암호문 E3을 인증 키 Kab로 복호화하고, 그 결과와 E2와의 배타적 논리합을 구하여 ID(b)를 얻는다. 이렇게 해서 얻어진 Ra, Rb, ID(b) 중, Rb 및 ID(b)가, B가 송신한 것과 일치하는지 검증한다. 이 검증에 통과한 경우, B는 A를 정당한 것으로 인증한다.

다음으로, B는 인증 후에 사용하는 세션 키(Kses)를 생성한다(생성 방법은 난수를 이용함). 그리고, Rb, Ra, Kses 순으로, DES의 CBC 모드로 인증 키 Kab를 이용하여 암호화하여, A에 반송한다.

이를 수신한 A는 수신 데이터를 인증 키 Kab로 복호화한다. 수신 데이터의 복호화 방법은 B의 복호화 처리와 마찬가지이므로, 여기서는 상세를 생략한다. 이렇게 해서 얻어진 Rb, Ra, Kses 중, Rb 및 Ra가, A가 송신한 것과 일치하는지 검증한다. 이 검증에 통과한 경우, A는 B를 정당한 것으로 인증한다. 상호 상대를 인증한 후에는 세션 키 Kses는 인증 후의 비밀 통신을 위한 공통 키로서 이용된다.

또, 수신 데이터의 검증 시, 부정, 불일치가 발견된 경우에는 상호 인증이 실패한 것으로 하여 처리를 중단한다.

상술한 인증 처리에 있어서는 A, B는 공통의 인증 키 Kab를 공유한다. 이 공통 키 Kab를 상술한 유효화 키 블록(EKB)을 사용하여 디바이스에 배신한다.

예를 들면, 도 12의 예에서는 A, 또는 B 중 어느 하나가 다른 쪽이 복호 가능한 유효화 키 블록(EKB)을 생성하여 생성한 유효화 키 블록(EKB)에 의해 인증 키 Kab를 암호화하여, 다른 쪽에 송신하는 구성으로 해도 되고, 또는 제3자가 디바이스 A, B에 대하여 양방이 이용 가능한 유효화 키 블록(EKB)을 생성하여 디바이스 A, B에 대하여 생성한 유효화 키 블록(EKB)에 의해 인증 키 Kab를 암호화하여 배신하는 구성으로 해도 무방하다.

도 13 및 도 14에 복수의 디바이스에 공통의 인증 키 Kake를 유효화 키 블록(EKB)에 의해 배신하는 구성예를 나타낸다. 도 13은 디바이스 0, 1, 2, 3에 대하여 복호 가능한 인증 키 Kake를 배신하는 예, 도 14는 디바이스 0, 1, 2, 3 중 디바이스 3을 리보크(배제)하여 디바이스 0, 1, 2에 대해서만 복호 가능한 인증 키를 배신하는 예를 나타낸다.

도 13의 예에서는 갱신 노드 키 K(t)00에 의해 인증 키 Kake를 암호화한 데이터와 함께, 디바이스 0, 1, 2, 3에 있어서 각각이 갖는 노드 키, 리프 키를 이용하여 갱신된 노드 키 K(t)00을 복호할 수 있는 유효화 키 블록(EKB)을 생성하여 배신한다. 각각의 디바이스는 도 13의 우측에 도시한 바와 같이 우선, EKB를 처리(복호)함으로써, 갱신된 노드 키 K(t)00을 취득하고, 이어서 취득한 노드 키 K(t)00을 이용하여 암호화된 인증 키: Enc(K(t)00, Kake)를 복호하여 인증 키 Kake를 얻을 수 있다.

그 밖의 디바이스 4, 5, 6, 7 …는 동일한 유효화 키 블록(EKB)을 수신해도 자신이 보유한 노드 키, 리프 키에서는 EKB를 처리하여 갱신된 노드 키 K(t)00을 취득할 수 없기 때문에, 안전하게 정당한 디바이스에 대해서만 인증 키를 송부할 수 있다.

한편, 도 14의 예는 도 3의 점선 프레임으로 둘러싼 그룹에서 디바이스 3이, 예를 들면 키 누설에 의해 리보크(배제)되어 있다고 하고, 다른 그룹의 멤버, 즉, 디바이스 0, 1, 2에 대해서만 복호 가능한 유효화 키 블록(EKB)을 생성하여 배신한 예이다. 도 14에 도시한 유효화 키 블록(EKB)과, 인증 키(Kake)를 노드 키(K(t)00)로 암호화한 데이터를 배신한다.

도 14의 우측에는 복호 순서를 나타내고 있다. 디바이스 0, 1, 2는 우선, 수령한 유효화 키 블록으로부터 자신이 보유한 리프 키 또는 노드 키를 이용한 복호 처리에 의해, 갱신 노드 키(K(t)00)를 취득한다. 다음으로, K(t)00에 의한 복호에 의해 인증 키 Kake를 취득한다.

도 3에 도시한 다른 그룹의 디바이스 4, 5, 6 …는 이와 유사한 데이터(EKB)를 수신했다고 해도, 자신이 보유한 리프 키, 노드 키를 이용하여 갱신 노드 키(K(t)00)를 취득할 수 없다. 마찬가지로 리보크된 디바이스 3에 있어서도, 자신이 보유한 리프 키, 노드 키에서는 갱신 노드 키(K(t)00)를 취득할 수 없고, 정당한 권리를 갖는 디바이스만이 인증 키를 복호하여 이용할 수 있다.

이와 같이 EKB를 이용한 인증 키의 배송을 이용하면, 데이터량을 적게 하고, 또한 안전하게 정당 권리자만이 복호 가능하게 한 인증 키를 배신할 수 있다.

[공개 키 인증과 유효화 키 블록(EKB)을 사용한 콘텐츠 키의 배신]

다음으로, 공개 키 인증과 유효화 키 블록(EKB)을 사용한 콘텐츠 키의 배신 처리에 대하여 설명한다. 우선, 공개 키 암호 방식인 160 비트 길이의 타원 곡선 암호를 이용한 상호 인증 방법에 대하여 도 15를 이용하여 설명한다. 도 15에서, 공개 키 암호 방식으로서 ECC를 이용하고 있지만, 마찬가지의 공개 키 암호 방식이면 어느 것이라도 무방하다. 또한, 키 사이즈도 160 비트가 아니어도 된다. 도 15에서, 우선 B가, 64 비트의 난수 Rb를 생성하여 A에 송신한다. 이를 수신한 A는 새롭게 64 비트의 난수 Ra 및 소수 p보다 작은 난수 Ak를 생성한다. 그리고, 베이스 포인트 G를 Ak배한 점 Av=Ak×G를 구하고, Ra, Rb, Av(X 좌표와 Y 좌표)에 대한 전자 서명 A.Sig를 생성하고, A의 공개 키 증명서와 함께 B에 반송한다. 여기서, Ra 및 Rb는 각각 64 비트, Av의 X 좌표와 Y 좌표가 각각 160 비트이기 때문에, 합계 448 비트에 대한 전자 서명을 생성한다.

A의 공개 키 증명서, Ra, Rb, Av, 전자 서명 A.Sig를 수신한 B는 A가 송신한 Rb가, B가 생성한 것과 일치하는지 검증한다. 그 결과, 일치한 경우에는 A의 공개 키 증명서 내의 전자 서명을 인증국의 공개 키로 검증하여, A의 공개 키를 추출한다. 그리고, 추출한 A의 공개 키를 이용하여 전자 서명 A.Sig를 검증한다.

다음으로, B는 소수 p보다 작은 난수 Bk를 생성한다. 그리고, 베이스 포인트 G를 Bk배한 점 Bv=Bk×G를 구하고, Rb, Ra, Bv(X 좌표와 Y 좌표)에 대한 전자 서명 B.Sig를 생성하고, B의 공개 키 증명서와 함께 A에 반송한다.

B의 공개 키 증명서, Rb, Ra, Av, 전자 서명 B.Sig를 수신한 A는 B가 송신한 Ra가, A가 생성한 것과 일치하는지 검증한다. 그 결과, 일치한 경우에는 B의 공개 키 증명서 내의 전자 서명을 인증국의 공개 키로 검증하여, B의 공개 키를 추출한다. 그리고, 추출한 B의 공개 키를 이용하여 전자 서명 B.Sig를 검증한다. 전자 서명의 검증에 성공한 후, A는 B를 정당한 것으로 인증한다.

양자가 인증에 성공한 경우에는 B는 Bk×Av(Bk는 난수이지만, Av는 타원 곡선 상의 점이기 때문에, 타원 곡선 상의 점의 스칼라배 계산이 필요)를 계산하고, A는 Ak×Bv를 계산하고, 이들 점의 X 좌표의 하위 64 비트를 세션 키로 하여 이후의 통신에 사용한다(공통 키 암호를 64 비트 키 길이의 공통 키 암호로 한 경우). 물론, Y 좌표부터 세션 키를 생성해도 되고, 하위 64 비트가 아니어도 된다. 또, 상호 인증 후의 비밀 통신에 있어서는 송신 데이터는 세션 키로 암호화될 뿐만 아니라, 전자 서명도 첨부되어 있는 경우가 있다.

전자 서명의 검증이나 수신 데이터의 검증 시, 부정, 불일치가 발견된 경우에는 상호 인증이 실패한 것으로 하여 처리를 중단한다.

도 16에 공개 키 인증과 유효화 키 블록(EKB)을 사용한 콘텐츠 키의 배신 처리예를 나타낸다. 우선, 콘텐츠 프로바이더와 PC 사이에서 도 15에서 설명한 공개 키 방식에 의한 인증 처리가 실행된다. 콘텐츠 프로바이더는 콘텐츠 키 배신처인 재생 장치, 기록 매체가 갖는 노드 키, 리프 키에 의해 복호 가능한 EKB를 생성하여, 갱신 노드 키에 의한 암호화를 실행한 콘텐츠 키 E(Kcon)와, 유효화 키 블록(EKB)을 PC 사이의 인증 처리에 있어서 생성한 세션 키 Kses로 암호화하여 PC에 송신한다.

PC는 세션 키로 암호화된 [갱신 노드 키에 의한 암호화를 실행한 콘텐츠 키 E(Kcon)와, 유효화 키 블록(EKB)]을 세션 키로 복호한 후, 재생 장치, 기록 매체에 송신한다.

재생 장치, 기록 매체는 자신이 보유한 노드 키 또는 리프 키에 의해 [갱신 노드 키에 의한 암호화를 실행한 콘텐츠 키 E(Kcon)와, 유효화 키 블록(EKB)]을 복호함으로써 콘텐츠 키 Kcon을 취득한다.

이 구성에 따르면, 콘텐츠 프로바이더와 PC 사이에서의 인증을 조건으로 하여 [갱신 노드 키에 의한 암호화를 실행한 콘텐츠 키 E(Kcon)와, 유효화 키 블록(EKB)]이 송신되기 때문에, 예를 들면, 노드 키의 누설이 있던 경우라도, 확실한 상대에 대한 데이터 송신이 가능하게 된다.

[프로그램 코드의 유효화 키 블록(EKB)을 사용한 배신]

상술한 예에서는 콘텐츠 키, 인증 키 등을 유효화 키 블록(EKB)을 이용하여 암호화하여 배신하는 방법을 설명했지만, 여러가지 프로그램 코드를 유효화 키 블록(EKB)을 이용하여 배신하는 구성도 가능하다. 즉, EKB에 의한 암호화 메시지 데이터를 프로그램 코드로 한 예이다. 이하, 이 구성에 대하여 설명한다.

도 17에 프로그램 코드를 유효화 키 블록(EKB)의, 예를 들면 갱신 노드 키에 의해 암호화하여 디바이스 사이에서 송신하는 예를 나타낸다. 디바이스(1701)는 디바이스(1702)가 갖는 노드 키, 리프 키에 의해 복호 가능한 유효화 키 블록(EKB)과, 유효화 키 블록(EKB)에 포함되는 갱신 노드 키로 암호 처리한 프로그램 코드를 디바이스(1702)에 송신한다. 디바이스(1702)는 수신한 EKB를 처리하여 갱신 노드 키를 취득하고, 또한 취득한 갱신 노드 키에 의해 프로그램 코드의 복호를 실행하여, 프로그램 코드를 얻는다.

도 17에 도시한 예에서는 또한, 디바이스(1702)에서 취득한 프로그램 코드에 의한 처리를 실행하여, 그 결과를 디바이스(1701)로 되돌리고, 디바이스(1701)가 그 결과에 기초하여 다시 처리를 속행하는 예를 나타내고 있다.

이와 같이 유효화 키 블록(EKB)과, 유효화 키 블록(EKB)에 포함되는 갱신 노드 키로 암호 처리한 프로그램 코드를 배신함으로써, 특정한 디바이스에 있어서 해독 가능한 프로그램 코드를 상술한 도 3에서 도시한 특정한 디바이스, 또는 그룹에 대하여 배신할 수 있다.

[송신 콘텐츠에 대한 체크치(ICV: Integrity Check Value)를 대응시키는 구성]

다음으로, 콘텐츠의 개찬을 방지하기 위해서 콘텐츠의 인터그리티·체크치 (ICV)를 생성하여, 콘텐츠에 대응하여, ICV의 계산에 의해 콘텐츠 개찬의 유무를 판정하는 처리 구성에 대하여 설명한다.

콘텐츠의 인터그리티·체크치(ICV)는, 예를 들면 콘텐츠에 대한 해시 함수를 이용하여 계산되고, ICV=hash(Kicv, C1, C2, …)에 의해 계산된다. Kicv는 ICV 생성 키이다. C1, C2는 콘텐츠의 정보이고, 콘텐츠의 중요 정보의 메시지 인증 부호(MAC: Message authentication Code)가 사용된다.

DES 암호 처리 구성을 이용한 MAC치 생성 예를 도 18에 도시한다. 도 18의 구성에 도시한 바와 같이 대상이 되는 메시지를 8바이트 단위로 분할하고(이하, 분할된 메시지를 M1, M2, …, MN으로 함), 우선, 초기치(Initial Value(이하, IV로 함))와 M1과의 배타적 논리합을 구한다(그 결과를 I1로 함). 다음으로, I1을 DES 암호화부에 넣어, 키(이하, K1로 함)를 이용하여 암호화한다(출력을 E1로 함). 계속해서, E1과 M2와의 배타적 논리합을 구하고, 그 출력 I2를 DES 암호화부에 넣어, 키 K1을 이용하여 암호화한다(출력 E2). 이하, 이를 반복하고, 모든 메시지에 대하여 암호화 처리를 실시한다. 마지막으로 출력된 EN이 메시지 인증 부호(MAC (Message Authentication Code))가 된다.

이러한 콘텐츠의 MAC치와 ICV 생성 키에 해시 함수를 적용하여 이용하여 콘텐츠의 인터그리티·체크치(ICV)가 생성된다. 개찬이 없는 것이 보증된, 예를 들면 콘텐츠 생성 시에 생성한 ICV와, 새롭게 콘텐츠에 기초하여 생성한 ICV를 비교하여 동일한 ICV가 얻어지면 콘텐츠에 개찬이 없는 것이 보증되고, ICV가 다르면, 개찬이 있었다고 판정된다.

[체크치(ICV)의 생성 키 Kicv를 EKB에 의해 배포하는 구성]

다음으로, 콘텐츠의 인터그리티·체크치(ICV) 생성 키인 Kicv를 상술한 유효화 키 블록에 의해 송부하는 구성에 대하여 설명한다. 즉, EKB에 의한 암호화 메시지 데이터를 콘텐츠의 인터그리티·체크치(ICV) 생성 키로 한 예이다.

도 19 및 도 20에 복수의 디바이스에 공통의 콘텐츠를 송부한 경우, 이들 콘텐츠의 개찬 유무를 검증하기 위한 인터그리티·체크치 생성 키 Kicv를 유효화 키 블록(EKB)에 의해 배신하는 구성예를 나타낸다. 도 19는 디바이스 0, 1, 2, 3에 대하여 복호 가능한 체크치 생성 키 Kicv를 배신하는 예, 도 20은 디바이스 0, 1, 2, 3 중의 디바이스 3을 리보크(배제)하여 디바이스 0, 1, 2에 대해서만 복호 가능한 체크치 생성 키 Kicv를 배신하는 예를 나타낸다.

도 19의 예에서는 갱신 노드 키 K(t)00에 의해 체크치 생성 키 Kicv를 암호화한 데이터와 함께, 디바이스 0, 1, 2, 3에 있어서 각각이 갖는 노드 키, 리프 키를 이용하여 갱신된 노드 키 K(t)00을 복호할 수 있는 유효화 키 블록(EKB)을 생성하여 배신한다. 각각의 디바이스는 도 19의 우측에 도시한 바와 같이 우선, EKB를 처리(복호)함으로써, 갱신된 노드 키 K(t)00을 취득하고, 이어서 취득한 노드 키 K(t)00을 이용하여 암호화된 체크치 생성 키: Enc(K(t)00, Kicv)를 복호하여 체크치 생성 키 Kicv를 얻을 수 있다.

그 밖의 디바이스 4, 5, 6, 7... 는 동일한 유효화 키 블록(EKB)을 수신해도 자신이 보유한 노드 키, 리프 키에서는 EKB를 처리하여 갱신된 노드 키 K(t)00을 취득할 수 없기 때문에, 안전하게 정당한 디바이스에 대해서만 체크치 생성 키를 송부할 수 있다.

한편, 도 20의 예는 도 3의 점선 프레임으로 둘러싼 그룹에서 디바이스 3이, 예를 들면 키 누설에 의해 리보크(배제)되어 있다고 해서, 다른 그룹의 멤버, 즉, 디바이스 0, 1, 2에 대해서만 복호 가능한 유효화 키 블록(EKB)을 생성하여 배신한 예이다. 도 20에 도시한 유효화 키 블록(EKB)과, 체크치 생성 키(Kicv)를 노드 키(K(t)00)로 암호화한 데이터를 배신한다.

도 20의 우측에는 복호 순서를 나타내고 있다. 디바이스 0, 1, 2는 우선, 수령한 유효화 키 블록으로부터 자신이 보유한 리프 키 또는 노드 키를 이용한 복호 처리에 의해, 갱신 노드 키(K(t)00)를 취득한다. 다음으로, K(t)00에 의한 복호에 의해 체크치 생성 키 Kicv를 취득한다.

도 3에 도시한 다른 그룹의 디바이스 4, 5, 6 …는 이와 유사한 데이터(EKB)를 수신했다고 해도, 자신이 보유한 리프 키, 노드 키를 이용하여 갱신 노드 키(K(t)00)를 취득할 수 없다. 마찬가지로 리보크된 디바이스 3에 있어서도, 자신이 보유한 리프 키, 노드 키로는 갱신 노드 키(K(t)00)를 취득할 수 없고, 정당한 권리를 갖는 디바이스만이 체크치 생성 키를 복호하여 이용할 수 있다.

이와 같이 EKB를 이용한 체크치 생성 키의 배송을 이용하면, 데이터량을 적게 하고, 또한 안전하게 정당 권리자만이 복호 가능하게 한 체크치 생성 키를 배신할 수 있다.

이러한 콘텐츠의 인터그리티·체크치(ICV)를 이용함으로써, EKB와 암호화 콘텐츠의 부정 복사를 배제할 수 있다. 예를 들면, 도 21A 및 도 21B에 도시한 바와 같이 콘텐츠 C1과 콘텐츠 C2를 각각의 콘텐츠 키를 취득 가능한 유효화 키 블록(EKB)과 함께 저장한 미디어 1이 있고, 이를 그대로 미디어 2에 복사한 경우를 상정한다. EKB와 암호화 콘텐츠의 복사는 가능하고, 이를 EKB를 복호할 수 있는 디바이스에서는 이용할 수 있게 된다.

도 21B에 도시한 바와 같이 각 미디어에 정당하게 저장된 콘텐츠에 대응하여 인터그리티·체크치(ICV(C1, C2))를 저장하는 구성으로 한다. 또, (ICV(C1, C2))는 콘텐츠 C1과 콘텐츠 C2에 해시 함수를 이용하여 계산되는 콘텐츠의 인터그리티·체크치인 ICV=hash(Kicv, C1, C2)를 나타내고 있다. 도 21B의 구성에서, 미디어 1에는 정당하게 콘텐츠 1과 콘텐츠 2가 저장되고, 콘텐츠 C1과 콘텐츠 C2에 기초하여 생성된 인터그리티·체크치(ICV(C1, C2))가 저장된다. 또한, 미디어 2에는 정당하게 콘텐츠 1이 저장되고, 콘텐츠 C1에 기초하여 생성된 인터그리티·체크치(ICV(C1))가 저장된다. 이 구성에서, 미디어 1에 저장된 {EKB, 콘텐츠 2}을 미디어 2에 복사했다고 하면, 미디어 2에서, 콘텐츠 체크치를 새롭게 생성하면 ICV(C1, C2)가 생성되게 되고, 미디어에 저장되어 있는 Kicv(C1)와 달리, 콘텐츠의 개찬 또는 부정 복사에 의한 새로운 콘텐츠의 저장이 실행된 것이 분명해진다. 미디어를 재생하는 디바이스에 있어서, 재생 단계의 이전 단계에 ICV 체크를 실행하여, 생성 ICV와 저장 ICV의 일치를 판별하고, 일치하지 않는 경우에는 재생을 실행하지 않은 구성으로 함으로써, 부정 복사의 콘텐츠의 재생을 방지할 수 있다.

또한, 안전성을 높이기 위해서, 콘텐츠의 인터그리티·체크치(ICV)를 재기입 카운터를 포함시킨 데이터에 기초하여 생성하는 구성으로 해도 무방하다. 즉, ICV=hash(Kicv, counter+1, C1, C2, …)에 의해 계산하는 구성으로 한다. 여기서, 카운터(counter+1)는 ICV의 재기입마다 하나 인크리먼트되는 값으로서 설정한다. 또, 카운터 값은 시큐어한 메모리에 저장하는 구성으로 할 필요가 있다.

또한, 콘텐츠의 인터그리티·체크치(ICV)를 콘텐츠와 동일 미디어에 저장할 수 없는 구성에서는 콘텐츠의 인터그리티·체크치(ICV)를 콘텐츠와는 다른 미디어 상에 저장하는 구성으로 해도 무방하다.

예를 들면, 판독 전용 미디어나 통상의 MO 등의 복사 방지책이 취해지고 있지 않은 미디어에 콘텐츠를 저장하는 경우, 동일 미디어에 인터그리티· 체크치(ICV)를 저장하면 ICV의 재기입이 부정한 사용자에 의해 이루어질 가능성이 있어, ICV의 안전성이 유지되지 않을 우려가 있다. 이러한 경우, 호스트 머신 상의 안전한 미디어에 ICV를 저장하여, 콘텐츠의 복사 컨트롤(예를 들면, check-in/check-out, move)에 ICV를 사용하는 구성으로 함으로써, ICV의 안전한 관리 및 콘텐츠의 개찬 체크가 가능하게 된다.

이 구성예를 도 22에 도시한다. 도 22에서는 판독 전용 미디어나 롬상의 MO 등의 복사 방지책이 취해지고 있지 않은 미디어(2201)에 콘텐츠가 저장되고, 이들 콘텐츠에 관한 인터그리티·체크치(ICV)를 사용자가 자유롭게 액세스하는 것이 허가되지 않은 호스트 머신 상의 안전한 미디어(2202)에 저장하고, 사용자에 의한 부정한 인터그리티·체크치(ICV)의 재기입을 방지한 예이다. 이러한 구성으로서, 예를 들면 미디어(2201)를 장착한 디바이스가 미디어(2201)의 재생을 실행할 때에 호스트 머신인 PC, 서버에 있어서 ICV의 체크를 실행하여 재생 가부를 판정하는 구성으로 하면, 부정 복사 콘텐츠 또는 개찬 콘텐츠의 재생을 방지할 수 있다.

[계층 트리 구조의 카테고리 분류]

암호 키를 루트 키, 노드 키, 리프 키 등, 도 3의 계층 트리 구조로서 구성하고, 콘텐츠 키, 인증 키, ICV 생성 키, 또는 프로그램 코드, 데이터 등을 유효화 키 블록(EKB)과 함께 암호화하여 배신하는 구성에 대하여 설명해 왔지만, 노드 키 등을 정의하고 있는 계층 트리 구조를 각 디바이스의 카테고리마다 분류하여 효율적인 키 갱신 처리, 암호화 키 배신, 데이터 배신을 실행하는 구성에 대하여, 이하 설명한다.

도 23에 계층 트리 구조의 카테고리의 분류의 일례를 나타낸다. 도 23에서, 계층 트리 구조의 최상단에는 루트 키 Kroot(2301)가 설정되고, 이하의 중간단에는 노드 키(2302)가 설정되고, 최하단에는 리프 키(2303)가 설정된다. 각 디바이스는 개개의 리프 키와, 리프 키로부터 루트 키에 이르는 일련의 노드 키, 루트 키를 보유한다.

여기서, 일례로서 최상단으로부터 제M단째의 임의의 노드를 카테고리 노드 (2304)로서 설정한다. 즉, 제M단째 노드의 각각을 특정 카테고리의 디바이스 설정 노드로 한다. 제M단의 하나의 노드를 정점으로 하여 이하, M+1단 이하의 노드, 리프는 그 카테고리에 포함되는 디바이스에 관한 노드 및 리프로 한다.

예를 들면, 도 23의 제M단째의 하나의 노드(2305)에는 카테고리[메모리 스틱(상표)]가 설정되고, 이 노드 이하에 연속한 노드, 리프는 메모리 스틱을 사용한 여러가지 디바이스를 포함하는 카테고리 전용의 노드 또는 리프로서 설정된다. 즉, 노드(2305) 이하를 메모리 스틱의 카테고리에 정의되는 디바이스의 관련 노드, 및 리프의 집합으로서 정의한다.

또한, M단으로부터 수단분 하위의 단을 서브 카테고리 노드(2306)로서 설정할 수 있다. 예를 들면, 도 23에 도시한 바와 같이 카테고리 [메모리 스틱] 노드(2305)의 2단 아래의 노드에, 메모리 스틱을 사용한 디바이스의 카테고리에 포함되는 서브 카테고리 노드로서, [재생 전용기]의 노드를 설정한다. 또한, 서브 카테고리 노드인 재생 전용기의 노드(2306) 이하에, 재생 전용기의 카테고리에 포함되는 음악 재생 기능이 부가된 전화의 노드(2307)가 설정되고, 그 하위에, 음악 재생 기능이 부가된 전화의 카테고리에 포함되는 [PHS] 노드(2308)와 [휴대 전화] 노드(2309)를 더 설정할 수 있다.

또한, 카테고리, 서브 카테고리는 디바이스 종류뿐만 아니라, 예를 들면 임의의 메이커, 콘텐츠 프로바이더, 결제 기관 등이 독자적으로 관리하는 노드, 즉 처리 단위, 관할 단위, 또는 제공 서비스 단위 등, 임의의 단위(이들을 총칭하여 이하, 엔티티라 함)로 설정할 수 있다. 예를 들면, 하나의 카테고리 노드를 게임 기기 메이커가 판매하는 게임 기기 XYZ 전용의 정점 노드로서 설정하면, 메이커가 판매하는 게임 기기 XYZ에 그 정점 노드 이하의 하단의 노드 키, 리프 키를 저장하여 판매할 수 있고, 그 후, 암호화 콘텐츠의 배신, 또는 각종 키의 배신, 갱신 처리를 그 정점 노드 키 이하의 노드 키, 리프 키에 의해 구성되는 유효화 키 블록(EKB)을 생성하여 배신하고, 정점 노드 이하의 디바이스에 대해서만 이용 가능한 데이터가 배신 가능하게 된다.

이와 같이 하나의 노드를 정점으로 하여, 이하의 노드를 그 정점 노드에 정의된 카테고리, 또는 서브 카테고리의 관련 노드로서 설정하는 구성으로 함으로써, 카테고리 단, 또는 서브 카테고리 단의 하나의 정점 노드를 관리하는 메이커, 콘텐츠 프로바이더 등이 그 노드를 정점으로 하는 유효화 키 블록(EKB)을 독자적으로 생성하여, 정점 노드 이하에 속하는 디바이스에 배신하는 구성이 가능하게 되고, 정점 노드에 속하지 않는 다른 카테고리의 노드에 속하는 디바이스에는 전혀 영향을 미치지 않고 키 갱신을 실행할 수 있다.

[간략 EKB에 의한 키 배신 구성]

먼저 설명한 예를 들면 도 3의 트리 구성에서, 키, 예를 들면 콘텐츠 키를 소정 디바이스(리프) 앞으로 송부하는 경우, 키 배포처 디바이스가 소유하고 있는 리프 키, 노드 키를 이용하여 복호 가능한 유효화 키 블록(EKB)을 생성하여 제공한다. 예를 들면, 도 24A에 도시한 트리 구성에서, 리프를 구성하는 디바이스 a, g, j에 대하여 키, 예를 들면 콘텐츠 키를 송신하는 경우, a, g, j의 각 노드에 있어서 복호 가능한 유효화 키 블록(EKB)을 생성하여 배신한다.

예를 들면, 갱신 루트 키 K(t)root로 콘텐츠 키 K(t)con을 암호화 처리하여, EKB와 함께 배신하는 경우를 생각한다. 이 경우, 디바이스 a, g, j는 각각이 도 24B에 도시한 리프 및 노드 키를 이용하여, EKB의 처리를 실행하여 K(t)root를 취득하고, 취득한 갱신 루트 키 K(t)root에 의해 콘텐츠 키 K(t)con의 복호 처리를 실행하여 콘텐츠 키를 얻는다.

이 경우에 제공되는 유효화 키 블록(EKB)의 구성은 도 25의 (A) 및 도 25의 (B)에 도시한 바와 같다. 도 25의 (A) 및 도 25의 (B)에 도시한 유효화 키 블록(EKB)은 앞의 도 6에서 설명한 유효화 키 블록(EKB)의 포맷에 따라 구성된 것으로, 데이터(암호화 키)와 대응하는 태그를 갖는다. 태그는, 먼저 도 7의 (A) 내지 도 7 (C)를 이용하여 설명한 바와 같이 좌측(L), 우측(R), 각각의 방향에 데이터가 있으면 0, 없으면 1을 나타내고 있다.

유효화 키 블록(EKB)을 수령한 디바이스는 유효화 키 블록(EKB)의 암호화 키와 태그에 기초하여 순차적으로 암호화 키의 복호 처리를 실행하여 상위 노드의 갱신 키를 취득한다. 도 25의 (A) 및 도 25의 (B)에 도시한 바와 같이 유효화 키 블록(EKB)은 루트에서 리프까지의 단 수(깊이)가 많을 수록, 그 데이터량은 증가한다. 단 수(깊이)는 디바이스(리프) 수에 따라 증대하는 것이고, 키의 배신처가 되는 디바이스 수가 많은 경우에는 EKB의 데이터량이 더욱 증대하게 된다.

이러한 유효화 키 블록(EKB)의 데이터량의 삭감을 가능하게 한 구성에 대하여 설명한다. 도 26의 (A) 및 도 26의 (B)는 유효화 키 블록(EKB)을 키 배신 디바이스에 따라 간략화하여 구성한 예를 나타내는 것이다.

도 25의 (A) 및 도 26의 (B)와 마찬가지로, 리프를 구성하는 디바이스 a, g, j에 대하여 키, 예를 들면 콘텐츠 키를 송신하는 경우를 상정한다. 도 26의 (A)에 도시한 바와 같이 키 배신 디바이스에 의해서만 구성되는 트리를 구축한다. 이 경우, 도 24B에 도시한 구성에 기초하여 새로운 트리 구성으로서 도 26의 (B)의 트리 구성이 구축된다. Kroot에서 Kj까지는 전혀 분기 없이 하나의 브랜치만이 존재하면 되고, Kroot에서 Ka 및 Kg에 이르기 위해서는 K0에 분기점을 구성하는 것만으로, 2분기 구성의 도 26의 (A)의 트리가 구축된다.

도 26의 (A)에 도시한 바와 같이 노드로서 K0만을 갖는 간략화한 트리가 생성된다. 갱신 키 배신을 위한 유효화 키 블록(EKB)은 이들 간략 트리에 기초하여 생성한다. 도 26의 (A)에 도시한 트리는 유효화 키 블록(EKB)을 복호할 수 있는 말단 노드 또는 리프를 최하단으로 한 2분기형 트리를 구성하는 패스를 선택하여 불필요한 노드를 생략함으로써 재구축되는 재구축 계층 트리이다. 갱신 키 배신을 위한 유효화 키 블록(EKB)은 이 재구축 계층 트리의 노드 또는 리프에 대응하는 키에만 기초하여 구성된다.

앞의 도 25의 (A) 및 도 25의 (B)에서 설명한 유효화 키 블록(EKB)은 각 리프 a, g, j에서 Kroot에 이르기까지의 모든 키를 암호화한 데이터를 저장했지만, 간략화 EKB는 간략화한 트리를 구성하는 노드에 대해서만의 암호화 데이터를 저장한다. 도 26의 (B)에 도시한 바와 같이 태그는 3비트 구성을 갖는다. 제1 및 제2 비트는 도 25의 (A) 및 도 25의 (B)의 예와, 마찬가지의 의미를 갖고, 좌측(L), 우측(R), 각각의 방향에 데이터가 있으면 0, 없으면 1을 나타낸다. 제3번째 비트는 EKB 내에 암호화 키가 저장되어 있는지의 여부를 나타내기 위한 비트로서, 데이터가 저장되어 있는 경우에는 1, 데이터가 없는 경우에는 0으로서 설정된다.

데이터 통신망, 또는 기억 매체에 저장되어 디바이스(리프)에 제공되는 유효화 키 블록(EKB)은 도 26의 (B)에 도시한 바와 같이 도 25의 (A) 및 도 25의 (B)에 도시한 구성에 비하면, 데이터량이 대폭 삭감된 것이 된다. 도 26의 (A) 및 도 26의 (B)에 도시한 유효화 키 블록(EKB)을 수령한 각 디바이스는 태그의 제3 비트에 1이 저장된 부분의 데이터만을 순차적으로 복호함으로써, 소정의 암호화 키의 복호를 실현할 수 있다. 예를 들면, 디바이스 a는 암호화 데이터 Enc(Ka, K(t)0)를 리프 키 Ka로 복호하여, 노드 키 K(t)0을 취득하여, 노드 키 K(t)0에 의해 암호화 데이터 Enc(K(t)0, K(t)root)를 복호하여 K(t)root를 취득한다. 디바이스 j는 암호화 데이터 Enc(Kj, K(t)root)를 리프 키 Kj로 복호하여 K(t)root를 취득한다.

이와 같이 배신처의 디바이스에 의해서만 구성되는 간략화한 새로운 트리 구성을 구축하여, 구축된 트리를 구성하는 리프 및 노드 키만을 이용하여 유효화 키 블록(EKB)을 생성함으로써, 적은 데이터량의 유효화 키 블록(EKB)을 생성할 수 있고, 유효화 키 블록(EKB)의 데이터 배신을 효율적으로 실행할 수 있다.

또, 간략화한 계층 트리 구성은 후단에서 설명하는 엔티티 단위의 EKB 관리 구성에서 특히 유효하게 활용 가능하다. 엔티티는 키 배신 구성으로서의 트리 구성을 구성하는 노드 또는 리프로부터 선택한 복수의 노드 또는 리프의 집합체 블록이다. 엔티티는 디바이스의 종류에 따라 설정되는 집합이거나, 또는 디바이스 제공 메이커, 콘텐츠 프로바이더, 결제 기관 등의 관리 단위 등, 어떤 공통점을 갖는 처리 단위, 관할 단위, 또는 제공 서비스 단위 등, 여러가지 양태의 집합으로서 설정된다. 하나의 엔티티에는 어떤 공통의 카테고리로 분류되는 디바이스가 모여 있으며, 예를 들면 복수의 엔티티의 정점 노드(서브 루트)에 의해 상술한 바와 마찬가지의 간략화한 트리를 재구축하여 EKB를 생성함으로써, 선택된 엔티티에 속하는 디바이스에 있어서 복호 가능한 간략화된 유효화 키 블록(EKB)의 생성, 배신이 가능하게 된다. 엔티티 단위의 관리 구성에 대해서는 후단에서 상세하게 설명한다.

또, 이러한 유효화 키 블록(EKB)은 광 디스크, DVD 등의 정보 기록 매체에 저장한 구성으로 할 수 있다. 예를 들면, 상술한 암호화 키 데이터에 의해 구성되는 데이터부와, 암호화 키 데이터의 계층 트리 구조에 있어서의 위치 식별 데이터로서의 태그부를 포함하는 유효화 키 블록(EKB)에 또한, 갱신 노드 키에 의해 암호화한 콘텐츠 등의 메시지 데이터를 저장한 정보 기록 매체를 각 디바이스에 제공하는 구성이 가능하다. 디바이스는 유효화 키 블록(EKB)에 포함되는 암호화 키 데이터를 태그부의 식별 데이터에 따라 순차적으로 추출하여 복호하고, 콘텐츠 복호에 필요한 키를 취득하여 콘텐츠의 이용을 행할 수 있다. 물론, 유효화 키 블록(EKB)을 인터넷 등의 네트워크를 통해 배신하는 구성으로 해도 무방하다.

[엔티티 단위의 EKB 관리 구성]

다음으로, 키 배신 구성으로서의 트리 구성을 구성하는 노드 또는 리프를 복수의 노드 또는 리프의 집합으로서의 블록으로 관리하는 구성에 대하여 설명한다. 또, 복수의 노드 또는 리프의 집합으로서의 블록을 이하 엔티티라고 한다. 엔티티는 디바이스의 종류에 따라 설정되는 집합이거나, 또는 디바이스 제공 메이커, 콘텐츠 프로바이더, 결제 기관 등의 관리 단위 등, 어떤 공통점을 갖는 처리 단위, 관할 단위, 또는 제공 서비스 단위 등, 여러가지 양태의 집합으로서 설정된다. 즉, 엔티티는 디바이스 종류, 서비스 종류, 관리 수단 종류 등의 공통의 카테고리에 속하는 디바이스 또는 엔티티의 관리 주체로서 정의된다.

엔티티에 대하여, 도 27의 (A) 내지 도 27 (C)를 이용하여 설명한다. 도 27의 (A)는 트리의 엔티티 단위의 관리 구성의 설명도이다. 하나의 엔티티는 도면에서는 상각형으로서 나타내고, 예를 들면 1 엔티티(2701) 내에는 복수의 노드가 포함된다. 1 엔티티 내의 노드 구성을 나타내는 것이 도 27의 (B)이다. 하나의 엔티티는 하나의 노드를 정점으로 한 복수단의 2분기형 트리에 의해 구성된다. 이하, 엔티티의 정점 노드(2702)를 서브 루트라 한다.

트리의 말단은 도 27 (C)에 도시한 바와 같이 리프, 즉 디바이스에 의해 구성된다. 디바이스는 복수 디바이스를 리프로 하고, 서브 루트인 정점 노드(2702)를 갖는 트리에 의해 구성되는 어느 하나의 엔티티에 속한다.

도 27의 (A)에서 알 수 있는 바와 같이, 엔티티는 계층 구조를 갖는다. 이 계층 구조에 대하여, 도 28의 (A) 내지 도 28 (C)를 이용하여 설명한다.

도 28의 (A)는 계층 구조를 간략화하여 설명하기 위한 도면이고, Kroot에서 수단 아래의 단에 엔티티 A01～Ann이 구성되고, 또한 엔티티 A1～An의 하위에는, 엔티티 B01～Bnk, 그리고, 그 하위에 엔티티 C1～Cnq가 설정되어 있다. 각 엔티티는 도 28의 (B), 도 28 (C)에 도시한 바와 같이 복수단의 노드, 리프에 의해 구성되는 트리 형상을 갖는다.

예를 들면, 엔티티 Bnk의 구성은 도 28의 (B)에 도시한 바와 같이 서브 루트(2811)를 정점 노드로서, 말단 노드(2812)에 이르기까지의 복수 노드를 갖는다. 이 엔티티는 식별자 Bnk를 갖고, 엔티티 Bnk 내의 노드에 대응하는 노드 키 관리를 엔티티 Bnk가 독자적으로 실행함으로써, 말단 노드(2812)를 정점으로 하여 설정되는 하위(종) 엔티티의 관리를 실행한다. 또한, 한편, 엔티티 Bnk는 서브 루트(2811)를 말단 노드로서 갖는 상위(주) 엔티티 Ann의 관리 하에 있다.

엔티티 Cn3의 구성은 도 28 (C)에 도시한 바와 같이 서브 루트(2851)를 정점 노드로 하여, 각 디바이스인 말단 노드(2852), 이 경우에는 리프에 이를 때까지 복수 노드, 리프를 갖는다. 이 엔티티는 식별자 Cn3을 갖고, 엔티티 Cn3 내의 노드, 리프에 대응하는 노드 키, 리프 키 관리를 엔티티 Cn3이 독자적으로 실행함으로써, 말단 노드(2852)에 대응하는 리프(디바이스)의 관리를 실행한다. 또한, 한편, 엔티티 Cn3은 서브 루트(2851)를 말단 노드로서 갖는 상위(주) 엔티티 Bn2의 관리 하에 있다. 각 엔티티에 있어서의 키 관리는, 예를 들면 키 갱신 처리, 리보크 처리 등이지만, 이들은 후단에서 상세히 설명한다.

최하단 엔티티의 리프인 디바이스에는 디바이스가 속하는 엔티티의 리프 키로부터, 자신이 속하는 엔티티의 정점 노드인 서브 루트 노드에 이르는 패스에 위치하는 각 노드의 노드 키 및 리프 키가 저장된다. 예를 들면, 말단 노드(2852)의 디바이스는 말단 노드(리프: 2852)로부터, 서브 루트 노드(2851)까지의 각 키를 저장한다.

도 29 (A) 및 도 29 (B)를 이용하여, 또한 엔티티의 구성에 대하여 설명한다. 엔티티는 여러가지 단 수에 의해 구성되는 트리 구조를 갖을 수 있다. 단 수, 즉 깊이(depth)는 엔티티로 관리하는 말단 노드에 대응하는 하위(종) 엔티티의 수, 또는 리프로서의 디바이스 수에 따라 설정 가능하다.

도 29 (A)에 도시한 바와 같은 상하 엔티티 구성을 구체화하면, 도 29 (B)에 도시한 양태가 된다. 루트 엔티티는 루트 키를 갖는 최상단의 엔티티이다. 루트 엔티티의 말단 노드에 복수의 하위 엔티티로서 엔티티 A, B, C가 설정되고, 또한 엔티티 C의 하위 엔티티로서 엔티티 D가 설정된다. 엔티티 C(2901)는 그 말단 노드의 하나 이상의 노드를 리저브 노드(2950)로서 보유하고, 자신이 관리하는 엔티티를 증가시키는 경우, 복수단의 트리 구성을 갖는 엔티티 C'(2902)를 리저브 노드(2950)를 정점 노드로서 신설함으로써, 관리 말단 노드(2970)를 증가시켜, 관리 말단 노드에 증가한 하위 엔티티를 더 추가할 수 있다.

리저브 노드에 대하여, 또한 도 30을 이용하여 설명한다. 엔티티 A(3011)는 관리하는 하위 엔티티 B, C, D …를 갖고, 하나의 리저브 노드(3021)를 갖는다. 엔티티는 관리 대상의 하위 엔티티를 더욱 증가시키고자 하는 경우, 리저브 노드(3021)에, 자기 관리의 하위 엔티티 A'(3012)를 설정하고, 하위 엔티티 A'(3012)의 말단 노드에 또한 관리 대상의 하위 엔티티 F, G를 설정할 수 있다. 자기 관리의 하위 엔티티 A'(3012)도, 그 말단 노드 중 적어도 하나를 리저브 노드(3022)로서 설정함으로써, 하위 엔티티 A"(3013)를 다시 설정하고, 관리 엔티티를 더욱 증가시킬 수 있다. 하위 엔티티 A"(3013)의 말단 노드에도 1이상의 리저브 노드를 확보한다. 이러한 리저브 노드 보유 구성을 취함으로써, 임의의 엔티티가 관리하는 하위 엔티티는 무한하게 증가시킬 수 있다. 또, 리저브 엔티티는 말단 노드의 하나만이 아니고, 복수개 설정하는 구성으로 해도 무방하다.

각각의 엔티티에서는 엔티티 단위로 유효화 키 블록(EKB)이 구성되고, 엔티티 단위의 키 갱신, 리보크 처리를 실행하게 된다. 도 30과 같이 복수의 엔티티 A, A', A"에는 각 엔티티 개개의 유효화 키 블록(EKB)이 설정되는 것으로 되지만, 이들은 엔티티 A, A', A"를 공통으로 관리하는, 예를 들면 임의의 디바이스 메이커가 일괄적으로 관리할 수 있다.

[신규 엔티티의 등록 처리]

다음으로, 신규 엔티티의 등록 처리에 대하여 설명한다. 등록 처리 시퀀스를 도 31에 도시한다. 도 31의 시퀀스에 따라 설명한다. 새롭게 트리 구성 중에 추가되는 신규(종) 엔티티(N-En)는 상위(주) 엔티티(P -En)에 대하여 신규 등록 요구를 실행한다. 또, 각 엔티티는 공개 키 암호 방식에 따른 공개 키를 보유하고, 신규 엔티티는 자신의 공개 키를 등록 요구에 있어서 상위 엔티티(P-En)에 송부한다.

등록 요구를 수령한 상위 엔티티(P-En)는 수령한 신규(종) 엔티티(N-En)의 공개 키를 증명서 발행국(CA: Certificate Authority)에 전송하고, CA의 서명을 부가한 신규(종) 엔티티(N-En)의 공개 키를 수령한다. 이들 수속은 상위 엔티티(P-En)와 신규(종) 엔티티(N-En)와의 상호 인증의 수속으로서 행해진다.

이들 처리에 의해, 신규 등록 요구 엔티티의 인증이 종료하면, 상위 엔티티 (P-En)는 신규(종) 엔티티(N-En)의 등록을 허가하고, 신규(종) 엔티티(N-En)의 노드 키를 신규(종) 엔티티(N-En)에 송신한다. 이 노드 키는 상위 엔티티(P-En)의 말단 노드의 하나의 노드 키로서, 또한 신규(종) 엔티티(N-En)의 정점 노드, 즉 서브 루트 키에 대응한다.

이 노드 키 송신이 종료하면, 신규(종) 엔티티(N-En)는 신규(종) 엔티티(N-En)의 트리 구성을 구축하고, 구축한 트리의 정점에 수신한 정점 노드의 서브 루트 키를 설정하고, 각 노드, 리프의 키를 설정하여, 엔티티 내의 유효화 키 블록(EKB)을 생성한다. 하나의 엔티티 내의 유효화 키 블록(EKB)을 서브 EKB라 한다.

한편, 상위 엔티티(P-En)는 신규(종) 엔티티(N-En)의 추가에 의해, 유효화하는 말단 노드를 추가한 상위 엔티티(P-En) 내의 서브 EKB를 생성한다.

신규(종) 엔티티(N-En)는 신규(종) 엔티티(N-En) 내의 노드 키, 리프 키에 의해 구성되는 서브 EKB를 생성하면, 이를 상위 엔티티(P-En)에 송신한다.

신규(종) 엔티티(N-En)로부터 서브 EKB를 수신한 상위 엔티티(P-En)는 수신한 서브 EKB와, 상위 엔티티(P-En)가 갱신한 서브 EKB를 키 발행 센터(KDC: Key Distribute Center)에 송신한다.

키 발행 센터(KDC)는 모든 엔티티의 서브 EKB에 기초하여 여러가지 양태의 EKB, 즉 특정한 엔티티 또는 디바이스만이 복호 가능한 EKB를 생성할 수 있다. 이와 같이 복호 가능한 엔티티 또는 디바이스를 설정한 EKB를, 예를 들면 콘텐츠 프로바이더에 제공하고, 콘텐츠 프로바이더가 EKB에 기초하여 콘텐츠 키를 암호화하여, 네트워크를 통해, 또는 기록 매체에 저장하여 제공함으로써, 특정한 디바이스에서만 이용 가능한 콘텐츠를 제공할 수 있다.

또, 신규 엔티티의 서브 EKB의 키 발행 센터(KDC)에 대한 등록 처리는 서브 EKB를 상위 엔티티를 통해 순차적으로 전송하여 실행하는 방법에 한하는 것이 아니고, 상위 엔티티를 통하지 않고, 신규 등록 엔티티로부터 직접, 키 발행 센터(KDC)에 등록하는 처리를 실행하는 구성으로 해도 무방하다.

상위 엔티티와, 상위 엔티티에 신규 추가하는 하위 엔티티와의 대응에 대하여 도 32를 이용하여 설명한다. 상위 엔티티의 하나의 말단 노드(3201)를 신규 추가 엔티티의 정점 노드로서, 하위 엔티티에 제공함으로써 하위 엔티티는 상위 엔티티의 관리하의 엔티티로서 추가된다. 상위 엔티티의 관리하의 엔티티는, 후단에서 상세하게 설명하지만, 하위 엔티티의 리보크(배제) 처리를 상위 엔티티가 실행할 수 있는 구성이라는 의미를 포함하는 것이다.

도 32에 도시한 바와 같이 상위 엔티티에 신규 엔티티가 설정되면, 상위 엔티티의 리프인 하나의 말단 노드(3201)와 신규 추가 엔티티의 정점 노드(3202)가 같은 노드로서 설정된다. 즉, 상위 노드의 하나의 리프인 하나의 말단 노드가 신규 추가 엔티티의 서브 루트로서 설정된다. 이와 같이 설정됨으로써, 신규 추가 엔티티가 전체 트리 구성 하에서 유효화된다.

도 33A 및 도 33B에 신규 추가 엔티티를 설정했을 때 상위 엔티티가 생성하는 갱신 EKB의 예를 나타낸다. 도 33A 및 도 33B는 도 33A에 도시한 구성, 즉 이미 유효하게 존재하는 말단 노드(node 000: 3301)와 말단 노드(node 001: 3302)가 있어, 여기에 신규 추가 엔티티에 신규 엔티티 추가 말단 노드(node 100: 3303)를 부여했을 때 상위 엔티티가 생성하는 서브 EKB의 예를 나타낸 것이다.

서브 EKB는 도 33B에 도시한 바와 같은 구성을 갖는다. 각각 유효하게 존재하는 말단 노드 키에 의해 암호화된 상위 노드 키, 상위 노드 키로 암호화된 상위 노드 키, … 또한 상위로 진행하여 서브 루트 키에 이르는 구성으로 되어 있다. 이 구성에 의해 서브 EKB가 생성된다. 각 엔티티는 도 33B에 도시한 바와 마찬가지로, 유효한 말단 노드, 또는 리프 키에 의해 암호화된 상위 노드 키, 상위 노드 키로 상위의 노드 키를 더욱 암호화하고, 순차적으로 상위에 심후하여 서브 루트에 이르는 암호화 데이터에 의해 구성되는 EKB를 갖고, 이를 관리한다.

[엔티티 관리 하에서의 리보크 처리]

다음으로, 키 배신 트리 구성을 엔티티 단위로서 관리하는 구성에서의 디바이스 또는 엔티티의 리보크(배제) 처리에 대하여 설명한다. 앞의 도 3, 4에서는 트리 구성 전체 중에서 특정한 디바이스만 복호 가능하고, 리보크된 디바이스는 복호 불가능한 유효화 키 블록(EKB)을 배신하는 처리에 대하여 설명하였다. 도 3, 도 4A 및 도 4B에서 설명한 리보크 처리는 트리 전체 중에서 특정한 리프인 디바이스를 리보크하는 처리였지만, 트리의 엔티티 관리에 의한 구성에서는 엔티티마다 리보크 처리를 실행할 수 있다.

도 34의 (A) 내지 도 34의 (D) 이하의 도면을 이용하여 엔티티 관리 하의 트리 구성에서의 리보크 처리에 대하여 설명한다. 도 34의 (A) 내지 도 34의 (D)는 트리를 구성하는 엔티티 중, 최하단의 엔티티, 즉 개개의 디바이스를 관리하고 있는 엔티티에 의한 디바이스의 리보크 처리의 설명도이다.

도 34의 (A)는 엔티티 관리에 의한 키 배신 트리 구조를 나타내고 있다. 트리 최상위에는 루트 노드가 설정되고, 그 수 단 아래에 엔티티 A01～Ann, 또한 그 하위단에 B01～Bnk의 엔티티, 또한 그 하위단에 C1～Cn의 엔티티가 구성되어 있다. 가장 아래의 엔티티는, 말단 노드(리프)가 개개의 디바이스, 예를 들면 기록 재생기, 재생 전용기 등이라고 한다.

여기서, 리보크 처리는 각 엔티티에 있어서 독자적으로 실행된다. 예를 들면, 최하단의 엔티티 C1～Cn에서는 리프의 디바이스의 리보크 처리가 실행된다. 도 34의 (B)에는 최하단의 엔티티의 하나인 엔티티 Cn(3430)의 트리 구성을 나타내고 있다. 엔티티 Cn(3430)은 정점 노드(3431)를 갖고, 말단 노드인 리프에 복수의 디바이스를 갖는 구성이다.

이 말단 노드인 리프 중에, 리보크 대상이 되는 디바이스, 예를 들면 디바이스(3432)가 있었다고 하면, 엔티티 Cn(3430)은 독자적으로 갱신한 엔티티 Cn 내의 노드 키, 리프 키에 의해 구성되는 유효화 키 블록(서브 EKB)을 생성한다. 이 유효화 키 블록은 리보크 디바이스(3432)에 있어서는 복호할 수 없고, 다른 리프를 구성하는 디바이스에 있어서만 복호 가능한 암호화 키에 의해 구성되는 키 블록이다. 엔티티 Cn의 관리자는 이를 갱신된 서브 EKB로서 생성한다. 구체적으로는 서브 루트에서 리보크 디바이스(3432)에 연속한 패스를 구성하는 각 노드(3431, 3434, 3435)의 노드 키를 갱신하고, 이 갱신 노드 키를 리보크 디바이스(3432) 이외의 리프 디바이스에 있어서만 복호 가능한 암호화 키로서 구성한 블록을 갱신 서브 EKB로 한다. 이 처리는 앞의 도 3, 4 (A) 및 도 4B에서 설명한 리보크 처리 구성에 있어서, 루트 키를 엔티티의 정점 키인 서브 루트 키로 치환한 처리에 대응한다.

이와 같이 엔티티 Cn(3430)이 리보크 처리에 의해 갱신한 유효화 키 블록(서브 EKB)은 상위 엔티티에 송신된다. 이 경우, 상위 엔티티는 엔티티 Bnk(3420)이고, 엔티티 Cn(3430)의 정점 노드(3431)를 말단 노드로서 갖는 엔티티이다.

엔티티 Bnk(3420)는 하위 엔티티 Cn(3430)으로부터 유효화 키 블록(서브 EKB)을 수령하면, 그 키 블록에 포함되는 엔티티 Cnk(3430)의 정점 노드(3431)에 대응하는 엔티티 Bnk(3420)의 말단 노드(3431)를 하위 엔티티 Cn(3430)에 있어서 갱신된 키로 설정하고, 자신의 엔티티 Bnk(3420)의 서브 EKB의 갱신 처리를 실행한다. 도 34의 (C)에 엔티티 Bnk(3420)의 트리 구성을 나타낸다. 엔티티 Bnk(3420)에 있어서, 갱신 대상이 되는 노드 키는 도 34의 (C)의 서브 루트(3421)로부터 리보크 디바이스를 포함하는 엔티티를 구성하는 말단 노드(3431)에 이르는 패스 상의 노드 키이다. 즉, 갱신 서브 EKB를 송신한 엔티티의 노드(3431)에 연속한 패스를 구성하는 각 노드(3421, 3424, 3425)의 노드 키가 갱신 대상이 된다. 이들 각 노드의 노드 키를 갱신하여 엔티티 Bnk(3420)가 새로운 갱신 서브 EKB를 생성한다.

또한, 엔티티 Bnk(3420)가 갱신한 유효화 키 블록(서브 EKB)은 상위 엔티티에 송신된다. 이 경우, 상위 엔티티는 엔티티 Ann(3410)으로서, 엔티티 Bnk(3420)의 정점 노드(3421)를 말단 노드로 갖는 엔티티이다.

엔티티 Ann(3410)은 하위 엔티티 Bnk(3420)로부터 유효화 키 블록(서브 EKB)을 수령하면, 그 키 블록에 포함되는 엔티티 Bnk(3420)의 정점 노드(3421)에 대응하는 엔티티 Ann(3410)의 말단 노드(3421)를 하위 엔티티 Bnk(3420)에 있어서 갱신된 키로 설정하여, 자신의 엔티티 Ann(3410)의 서브 EKB의 갱신 처리를 실행한다. 도 34의 (D)에 엔티티 Ann(3410)의 트리 구성을 나타낸다. 엔티티 Ann(3410)에 있어서, 갱신 대상이 되는 노드 키는 도 34의 (D)의 서브 루트(3411)로부터 갱신 서브 EKB를 송신한 엔티티의 노드(3421)에 연속해 있는 패스를 구성하는 각 노드(3411, 3414, 3415)의 노드 키이다. 이들 각 노드의 노드 키를 갱신하여 엔티티 Ann(3410)이 새로운 갱신 서브 EKB를 생성한다.

이들 처리를 순차적으로, 상위의 엔티티에 있어서 실행하고, 도 29 (B)에서 설명한 루트 엔티티까지 실행한다. 이 일련의 처리에 의해, 디바이스의 리보크 처리가 완결한다. 또, 각각의 엔티티에 있어서 갱신된 서브 EKB는 최종적으로 키 발행 센터(KDC)에 송신되어 보관된다. 키 발행 센터(KDC)는 모든 엔티티의 갱신 서브 EKB에 기초하여 여러가지 EKB를 생성한다. 갱신 EKB는 리보크된 디바이스에서의 복호가 불가능한 암호화 키 블록이 된다.

디바이스의 리보크 처리의 시퀀스도를 도 35에 도시한다. 처리 순서를 도 35의 시퀀스도에 따라 설명한다. 우선, 트리 구성의 최하단에 있는 디바이스 관리 엔티티(D-En)는 디바이스 관리 엔티티(D-En) 내의 리보크 대상의 리프를 배제하기 위해서 필요한 키 갱신을 행하고, 디바이스 관리 엔티티(D-En)가 새로운 서브 EKB(D)를 생성한다. 갱신 서브 EKB(D)는 상위 엔티티에 송부된다. 갱신 서브 EKB(D)를 수령한 상위(주) 엔티티(P1-En)는 갱신 서브 EKB(D)의 갱신 정점 노드에 대응한 말단 노드 키의 갱신 및, 그 말단 노드로부터 서브 루트에 이르는 패스 상의 노드 키를 갱신한 갱신 서브 EKB(P1)를 생성한다. 이들 처리를 순차적으로, 상위 엔티티에 있어서 실행하여, 최종적으로 갱신된 모든 서브 EKB가 키 발행 센터(KDC)에 저장되어 관리된다.

도 36A 및 도 36B에 디바이스의 리보크 처리에 의해 상위 엔티티가 갱신 처리를 행하여 생성하는 유효화 키 블록(EKB)의 예를 나타낸다.

도 36A 및 도 36B는 도 36A에 도시한 구성에서, 리보크 디바이스를 포함하는 하위 엔티티로부터 갱신 서브 EKB를 수신한 상위 엔티티에 있어서 생성하는 EKB의

예의 설명도이다. 리보크 디바이스를 포함하는 하위 엔티티의 정점 노드는 상위 엔티티의 말단 노드(node100: 3601)에 대응한다.

상위 엔티티는 상위 엔티티의 서브 루트에서 말단 노드(node100: 3601)까지의 패스에 존재하는 노드 키를 갱신하여 새로운 갱신 서브 EKB를 생성한다. 갱신 서브 EKB는 도 36B와 같이 된다. 갱신된 키는 하부선 및 [']을 붙여 나타내고 있다. 이와 같이 갱신된 말단 노드부터 서브 루트까지의 패스 상의 노드 키를 갱신하여 그 엔티티에 있어서의 갱신 서브 EKB로 한다.

다음으로, 리보크하는 대상을 엔티티로 한 경우의 처리, 즉 엔티티의 리보크 처리에 대하여 설명한다.

도 37의 (A)는 엔티티 관리에 의한 키 배신 트리 구조를 나타내고 있다. 트리 최상위에는 루트 노드가 설정되고, 그 수 단 아래에 엔티티 A01～Ann, 또한 그 하위단에 B01～Bnk의 엔티티, 또한 그 하위단에 C1～Cn의 엔티티가 구성되어 있다. 가장 아래의 엔티티는 말단 노드(리프)가 개개의 디바이스, 예를 들면 기록 재생기, 재생 전용기 등이라고 한다.

여기서, 리보크 처리를 엔티티 Cn(3730)에 대하여 실행하는 경우에 대해서 설명한다. 최하단의 엔티티 Cn(3730)은 도 37의 (B)에 도시한 바와 같이 정점 노드 (3431)를 갖고, 말단 노드인 리프에 복수의 디바이스를 갖는 구성이다.

엔티티 Cn(3730)을 리보크함으로써, 엔티티 Cn(3730)에 속하는 모든 디바이스의 트리 구조로부터의 일괄 배제가 가능하게 된다. 엔티티 Cn(3730)의 리보크 처리는 엔티티 Cn(3730)의 상위 엔티티인 엔티티 Bnk(3720)에 있어서 실행된다. 엔티티 Bnk(3720)는 엔티티 Cn(3730)의 정점 노드(3731)를 말단 노드로서 갖는 엔티티이다.

엔티티 Bnk(3720)는 하위 엔티티 Cn(3730)의 리보크를 실행하는 경우, 엔티티 Cnk(3730)의 정점 노드(3731)에 대응하는 엔티티 Bnk(3720)의 말단 노드(3731)를 갱신하고, 또한 그 리보크 엔티티(3730)로부터 엔티티 Bnk(3720)의 서브 루트까지의 패스 상의 노드 키의 갱신을 행하여 유효화 키 블록을 생성하여 갱신 서브 EKB를 생성한다. 갱신 대상이 되는 노드 키는 도 37 (C)의 서브 루트(3721)로부터 리보크 엔티티의 정점 노드를 구성하는 말단 노드(3731)에 이르는 패스 상의 노드 키이다. 즉, 노드(3721, 3724, 3725, 3731)의 노드 키가 갱신 대상이 된다. 이들 각 노드의 노드 키를 갱신하여 엔티티 Bnk(3720)가 새로운 갱신 서브 EKB를 생성한다.

또는 엔티티 Bnk(3720)는 하위 엔티티 Cn(3730)의 리보크를 실행하는 경우, 엔티티 Cnk(3730)의 정점 노드(3731)에 대응하는 엔티티 Bnk(3720)의 말단 노드(3731)는 갱신하지 않고, 그 리보크 엔티티(3730)로부터 엔티티 Bnk(3720)의 서브 루트까지의 패스 상의 말단 노드(3731)를 제외한 노드 키의 갱신을 행하여 유효화 키 블록을 생성하여 갱신 서브 EKB를 생성해도 된다.

또한, 엔티티 Bnk(3720)가 갱신한 유효화 키 블록(서브 EKB)은 상위 엔티티에 송신된다. 이 경우, 상위 엔티티는 엔티티 Ann(3710)으로서, 엔티티 Bnk(3720)의 정점 노드(3721)를 말단 노드로서 갖는 엔티티이다.

엔티티 Ann(3710)은 하위 엔티티 Bnk(3720)로부터 유효화 키 블록(서브 EKB)을 수령하면, 그 키 블록에 포함되는 엔티티 Bnk(3720)의 정점 노드(3721)에 대응하는 엔티티 Ann(3710)의 말단 노드(3721)를 하위 엔티티 Bnk(3720)에 있어서 갱신된 키로 설정하여, 자신의 엔티티 Ann(3710)의 서브 EKB의 갱신 처리를 실행한다. 도 37 (D)에 엔티티 Ann(3710)의 트리 구성을 나타낸다. 엔티티 Ann(3710)에 있어서, 갱신 대상이 되는 노드 키는 도 37 (D)의 서브 루트(3711)로부터 갱신 서브 EKB를 송신하는 엔티티의 노드(3721)에 연속한 패스를 구성하는 각 노드(3711, 3714, 3715)의 노드 키이다. 이들 각 노드의 노드 키를 갱신하여 엔티티 Ann(3710)이 새로운 갱신 서브 EKB를 생성한다.

이들 처리를 순차적으로, 상위의 엔티티에 있어서 실행하고, 도 29 (D)에서 설명한 루트 엔티티까지 실행한다. 이 일련의 처리에 의해, 엔티티의 리보크 처리가 완결한다. 또, 각각의 엔티티에 있어서 갱신된 서브 EKB는 최종적으로 키 발행 센터(KDC)에 송신되어 보관된다. 키 발행 센터(KDC)는 모든 엔티티의 갱신 서브 EKB에 기초하여 여러가지 EKB를 생성한다. 갱신 EKB는 리보크된 엔티티에 속하는 디바이스에서의 복호가 불가능한 암호화 키 블록이 된다.

엔티티의 리보크 처리의 시퀀스도를 도 38에 도시한다. 처리 순서를 도 38의 시퀀스도에 따라 설명한다. 우선, 엔티티를 리보크하고자 하는 엔티티 관리 엔티티(E-En)는 엔티티 관리 엔티티(E-En) 내의 리보크 대상의 말단 노드를 배제하기 위해서 필요한 키 갱신을 행하고, 엔티티 관리 엔티티(E-En)의 새로운 서브 EKB(E)를 생성한다. 갱신 서브 EKB(E)는 상위 엔티티에 송부된다. 갱신 서브 EKB(E)를 수령한 상위(주) 엔티티(P1-En)는 갱신 서브 EKB(E)의 갱신 정점 노드에 대응한 말단 노드 키의 갱신 및, 그 말단 노드로부터 서브 루트에 이르는 패스 상의 노드 키를 갱신한 갱신 서브 EKB(P1)를 생성한다. 이들 처리를 순차적으로, 상위 엔티티에 있어서 실행하여, 최종적으로 갱신된 모든 서브 EKB가 키 발행 센터(KDC)에 저장되어 관리된다. 키 발행 센터(KDC)는 모든 엔티티의 갱신 서브 EKB에 기초하여 여러가지 EKB를 생성한다. 갱신 EKB는 리보크된 엔티티에 속하는 디바이스에서의 복호가 불가능한 암호화 키 블록이 된다.

도 39에 리보크된 하위 엔티티와, 리보크를 행한 상위 엔티티의 대응을 설명하는 도면을 도시한다. 상위 엔티티의 말단 노드(3901)는 엔티티의 리보크에 의해 갱신되고, 상위 엔티티의 트리에 있어서의 말단 노드(3901)부터 서브 루트까지의 패스에 존재하는 노드 키의 갱신에 의해, 새로운 서브 EKB가 생성된다. 그 결과, 리보크된 하위 엔티티의 정점 노드(3902)의 노드 키와, 상위 엔티티의 말단 노드(3901)의 노드 키는 불일치가 된다. 엔티티의 리보크 후에 키 발행 센터(KDC)에 의해 생성되는 EKB는 상위 엔티티에 있어서 갱신된 말단 노드(3901)의 키에 기초하여 생성되므로, 그 갱신 키를 보유하지 않은 하위 엔티티의 리프에 대응하는 디바이스는 키 발행 센터(KDC)에 의해 생성되는 EKB의 복호가 불가능하게 된다.

또, 상술한 설명에서는 디바이스를 관리하는 최하단의 엔티티의 리보크 처리에 대하여 설명했지만, 트리의 중단에 있는 엔티티 관리 엔티티를 그 상위 엔티티가 리보크하는 처리도 상기와 마찬가지의 프로세스에 의해 가능하다. 중단의 엔티티 관리 엔티티를 리보크함으로써, 리보크된 엔티티 관리 엔티티의 하위에 속하는 모든 복수 엔티티 및 디바이스를 일괄적으로 리보크 가능하게 된다.

이와 같이 엔티티 단위의 리보크를 실행함으로써, 하나 하나의 디바이스 단위로 실행하는 리보크 처리에 비하여 간이한 프로세스에서의 리보크 처리가 가능하게 된다.

[엔티티의 캐이퍼빌리티 관리]

다음으로, 엔티티 단위의 키 배신 트리 구성에서, 각 엔티티가 허용하는 캐이퍼빌리티(Capability)를 관리하여, 캐이퍼빌리티에 따른 콘텐츠 배신을 행하는 처리 구성에 대하여 설명한다. 여기서 캐이퍼빌리티는, 예를 들면 특정한 압축 음성 데이터의 복호가 가능하거나, 특정한 음성 재생 방식을 허용하거나, 또는 특정한 화상 처리 프로그램을 처리할 수 있는 등, 디바이스가 어떠한 콘텐츠, 또는 프로그램 등을 처리할 수 있는 디바이스인지, 즉 디바이스의 데이터 처리 능력의 정의 정보이다.

도 40에 캐이퍼빌리티를 정의한 엔티티 구성예를 나타낸다. 키 배신 트리 구성의 최정점에 루트 노드가 위치하고, 하층에 복수의 엔티티가 접속되어 각 노드가 2분기를 갖는 트리 구성이다. 여기서, 예를 들면 엔티티(4001)는 음성 재생 방식 A, B, C 중 어느 하나를 허용하는 캐이퍼빌리티를 갖는 엔티티로서 정의된다. 구체적으로는 예를 들면 임의의 음성 압축 프로그램-A, B, 또는 C 방식으로 압축한 음악 데이터를 배신한 경우에, 엔티티(4001) 이하에 구성된 엔티티에 속하는 디바이스는 압축 데이터를 신장하는 처리가 가능하다.

마찬가지로, 엔티티(4002)는 음성 재생 방식 B 또는 C, 엔티티(4003)는 음성 재생 방식 A 또는 B, 엔티티(4004)는 음성 재생 방식 B, 엔티티(4005)는 음성 재생 방식 C를 처리할 수 있는 캐이퍼빌리티를 갖는 엔티티로서 정의된다.

한편, 엔티티(4021)는 화상 재생 방식 p, q, r을 허용하는 엔티티로서 정의되고, 엔티티(4022)는 방식 p, q의 화상 재생 방식, 엔티티(4023)는 방식 p의 화상 재생이 가능한 캐이퍼빌리티를 갖는 엔티티로서 정의된다.

이러한 각 엔티티의 캐이퍼빌리티 정보는 키 발행 센터(KDC)에서 관리된다. 키 발행 센터(KDC)는, 예를 들면 임의의 콘텐츠 프로바이더가 특정한 압축 프로그램으로 압축한 음악 데이터를 여러가지 디바이스에 배신하고자 하는 경우, 그 특정한 압축 프로그램을 재생 가능한 디바이스에 대해서만 복호할 수 있는 유효화 키 블록(EKB)을 각 엔티티의 캐이퍼빌리티 정보에 기초하여 생성할 수 있다. 콘텐츠를 제공하는 콘텐츠 프로바이더는 캐이퍼빌리티 정보에 기초하여 생성한 유효화 키 블록 (EKB)에 의해 암호화한 콘텐츠 키를 배신하고, 그 콘텐츠 키로 암호화한 압축 음성 데이터를 각 디바이스에 제공한다. 이 구성에 의해, 데이터 처리가 가능한 디바이스에 대해서만 특정한 처리 프로그램을 확실하게 제공할 수 있다 .

또, 도 40에서는 모든 엔티티에 대하여 캐이퍼빌리티 정보를 정의하고 있는 구성이지만, 도 40의 구성과 같이 모든 엔티티에 캐이퍼빌리티 정보를 정의하는 것은 반드시 필요한 것이 아니라, 예를 들면 도 41에 도시한 바와 같이 디바이스가 속하는 최하단의 엔티티에 대해서만 캐이퍼빌리티를 정의하고, 최하단의 엔티티에 속하는 디바이스의 캐이퍼빌리티를 키 발행 센터(KDC)에서 관리하여, 콘텐츠 프로바이더가 기대하는 처리가 가능한 디바이스에만 복호 가능한 유효화 키 블록(EKB)을 최하단의 엔티티에 정의된 캐이퍼빌리티 정보에 기초하여 생성하는 구성으로 해도 무방하다. 도 41에서는 말단 노드에 디바이스가 정의된 엔티티(4101=4105)에 있어서의 캐이퍼빌리티가 정의되고, 이들 엔티티에 대한 캐이퍼빌리티를 키 발행 센터(KDC)에서 관리하는 구성이다. 예를 들면, 엔티티(4101)에는 음성 재생에 대해서는 방식 B, 화상 재생에 대해서는 방식 r의 처리가 가능한 디바이스가 속해 있다. 엔티티(4102)에는 음성 재생에 대해서는 방식 A, 화상 재생에 대해서는 방식 q의 처리가 가능한 디바이스가 속해 있다.

도 42의 (A) 및 도 42의 (B)에 키 발행 센터(KDC)에서 관리하는 캐이퍼빌리티 관리 테이블의 구성예를 나타낸다. 캐이퍼빌리티 관리 테이블은 도 42의 (A)와 같은 데이터 구성을 갖는다. 즉, 각 엔티티를 식별하는 식별자로서의 엔티티 ID, 그 엔티티에 정의된 캐이퍼빌리티를 나타내는 캐이퍼빌리티 리스트, 이 캐이퍼빌리티 리스트는 도 42의 (B)에 도시한 바와 같이, 예를 들면 음성 데이터 재생 처리(방식 A)가 처리 가능하면 [1], 처리 불가능하면 [0], 음성 데이터 재생 처리(방식 B)가 처리 가능하면 [1], 처리 불가능하면 [0] … 등, 여러가지 양태의 데이터 처리에 대한 가부를 1 비트씩 [1] 또는 [0]을 설정하여 구성되어 있다. 또, 이 캐이퍼빌리티 정보의 설정 방법은 이러한 형식에 한하지 않고, 엔티티의 관리 디바이스에 대한 캐이퍼빌리티를 식별 가능하면 다른 구성이라도 무방하다.

캐이퍼빌리티 관리 테이블에는 또한, 각 엔티티의 서브 EKB, 또는 서브 EKB가 다른 데이터 베이스에 저장되어 있는 경우에는 서브 EKB의 식별 정보가 저장되고, 또한 각 엔티티의 서브 루트 노드 식별 데이터가 저장된다.

키 발행 센터(KDC)는 캐이퍼빌리티 관리 테이블에 기초하여, 예를 들면 특정한 콘텐츠의 재생 가능한 디바이스만이 복호 가능한 유효화 키 블록(EKB)을 생성한다. 도 43을 이용하여, 캐이퍼빌리티 정보에 기초한 유효화 키 블록의 생성 처리에 대하여 설명한다.

우선, 단계 S4301에서, 키 발행 센터(KDC)는 캐이퍼빌리티 관리 테이블로부터, 지정된 캐이퍼빌리티를 갖는 엔티티를 선택한다. 구체적으로는, 예를 들면 콘텐츠 프로바이더가 음성 데이터 재생 처리 방식 A에 기초한 재생 가능한 데이터를 배신하고자 하는 경우에는 도 42의 (A)의 캐이퍼빌리티 리스트로부터, 예를 들면 음성 데이터 재생 처리(방식 A)의 항목이 [1]로 설정된 엔티티를 선택한다.

다음으로, 단계 S4302에서, 선택된 엔티티에 의해 구성되는 선택 엔티티 ID의 리스트를 생성한다. 다음으로, 단계 S4303에서, 선택 엔티티 ID에 의해 구성되는 트리에 필요한 패스(키 배신 트리 구성의 패스)를 선택한다. 단계 S4304에서는 선택 엔티티 ID의 리스트에 포함되는 모든 패스 선택이 완료했는지의 여부를 판정하고, 완료할 때까지, 단계 S4303에서 패스를 생성한다. 이는 복수의 엔티티가 선택된 경우에, 각각의 패스를 순차적으로 선택하는 처리를 의미하고 있다.

선택 엔티티 ID의 리스트에 포함되는 모든 패스 선택이 완료하면, 단계 S4305로 진행하고, 선택한 패스와, 선택 엔티티에 의해서만 구성되는 키 배신 트리 구조를 구축한다.

다음으로, 단계 S4306에서, 단계 S4305에서 생성한 트리 구조의 노드 키의 갱신 처리를 행하여, 갱신 노드 키를 생성한다. 또한, 트리를 구성하는 선택 엔티티의 서브 EKB를 캐이퍼빌리티 관리 테이블로부터 추출하고, 서브 EKB와, 단계 S4306에서 생성한 갱신 노드 키에 기초하여 선택 엔티티의 디바이스에 있어서만 복호 가능한 유효화 키 블록(EKB)을 생성한다. 이와 같이 하여 생성한 유효화 키 블록(EKB)은 특정한 캐이퍼빌리티를 갖는 디바이스에 있어서만 이용, 즉 복호 가능한 유효화 키 블록(EKB)이 된다. 이 유효화 키 블록(EKB)으로, 예를 들면 콘텐츠 키를 암호화하여, 그 콘텐츠 키로 특정 프로그램에 기초하여 압축한 콘텐츠를 암호화하여 디바이스에 제공함으로써, 키 발행 센터(KDC)에 의해 선택된 특정한 처리 가능한 디바이스에 있어서만 콘텐츠가 이용된다.

이와 같이 키 발행 센터(KDC)는 캐이퍼빌리티 관리 테이블에 기초하여, 예를 들면 특정한 콘텐츠의 재생 가능한 디바이스만이 복호 가능한 유효화 키 블록(EKB)을 생성한다. 따라서, 새로운 엔티티가 등록되는 경우에는 그 신규 등록 엔티티의 캐이퍼빌리티를 사전에 취득할 필요가 있다. 이 엔티티 신규 등록에 따른 캐이퍼빌리티 통지 처리에 대하여 도 44를 이용하여 설명한다.

도 44는 신규 엔티티가 키 배신 트리 구성에 참가하는 경우의 캐이퍼빌리티 통지 처리 시퀀스를 나타낸 도면이다.

새롭게 트리 구성 중에 추가되는 신규(종) 엔티티(N-En)는 상위(주) 엔티티 (P-En)에 대하여 신규 등록 요구를 실행한다. 또, 각 엔티티는 공개 키 암호 방식에 따른 공개 키를 보유하고, 신규 엔티티는 자신의 공개 키를 등록 요구 시에, 상위 엔티티(P-En)에 송부한다.

등록 요구를 수령한 상위 엔티티(P-En)는 수령한 신규(종) 엔티티(N-En)의 공개 키를 증명서 발행국(CA: Certificate Authority)에 전송하고, CA의 서명을 부가한 신규(종) 엔티티(N-En)의 공개 키를 수령한다. 이들 수속은 상위 엔티티(P-En)와 신규(종) 엔티티(N-En)와의 상호 인증의 수속으로서 행해진다.

이들 처리에 의해, 신규 등록 요구 엔티티의 인증이 종료하면, 상위 엔티티 (P-En)는 신규(종) 엔티티(N-En)의 등록을 허가하고, 신규(종) 엔티티(N-En)의 노드 키를 신규(종) 엔티티(N-En)에 송신한다. 이 노드 키는 상위 엔티티(P-En)의 말단 노드의 하나의 노드 키로서, 또한 신규(종) 엔티티(N-En)의 정점 노드, 즉 서브 루트 키에 대응한다.

이 노드 키 송신이 종료하면, 신규(종) 엔티티(N-En)는 신규(종) 엔티티(N-En)의 트리 구성을 구축하고, 구축한 트리의 정점에 수신한 정점 노드의 서브 루트 키를 설정하고, 각 노드, 리프 키를 설정하여 엔티티 내의 유효화 키 블록(서브 EKB)을 생성한다. 한편, 상위 엔티티(P-En)도 신규(종) 엔티티(N-En)의 추가에 의해 유효화하는 말단 노드를 추가한 상위 엔티티(P-En) 내의 서브 EKB를 생성한다.

신규(종) 엔티티(N-En)는 신규(종) 엔티티(N-En) 내의 노드 키, 리프 키에 의해 구성되는 서브 EKB를 생성하면, 이를 상위 엔티티(P-En)에 송신하고, 또한 자신의 엔티티로 관리하는 디바이스에 대한 캐이퍼빌리티 정보를 상위 엔티티에 통지한다.

신규(종) 엔티티(N-En)로부터 서브 EKB 및 캐이퍼빌리티 정보를 수신한 상위 엔티티(P-En)는 수신한 서브 EKB와 캐이퍼빌리티 정보와, 상위 엔티티(P-En)가 갱신한 서브 EKB를 키 발행 센터(KDC: Key Distribute Center)에 송신한다.

키 발행 센터(KDC)는 수령한 엔티티의 서브 EKB 및 캐이퍼빌리티 정보를 도 42의 (A) 및 도 42의 (B)에서 설명한 캐이퍼빌리티 관리 테이블에 등록하고, 캐이퍼빌리티 관리 테이블을 갱신한다. 키 발행 센터(KDC)는 갱신한 캐이퍼빌리티 관리 테이블에 기초하여 여러가지 양태의 EKB, 즉 특정한 캐이퍼빌리티를 갖는 엔티티, 또는 디바이스만이 복호 가능한 EKB를 생성할 수 있다.

이상, 특정한 실시예를 참조하면서, 본 발명에 대하여 상세하게 설명하였다. 그러나, 본 발명의 요지를 일탈하지 않는 범위에서 당업자가 실시예의 수정이나 대용을 할 수 있는 것은 자명하다. 즉, 예시라는 형태로 본 발명을 개시하여 온 것으로, 한정적으로 해석되어서는 안 된다. 본 발명의 요지를 판단하기 위해서는 첫머리에 기재한 특허 청구의 범위의 란을 참작해야 한다.

산업상이용가능성

이상, 설명한 바와 같이 본 발명에 따른 암호 키 블록을 이용한 정보 처리 시스템 및 방법에 따르면, 복수의 디바이스를 리프로 구성한 트리의 루트에서 리프까지의 패스 상의 루트, 노드, 및 리프에 각각 키를 대응시킨 키 트리를 디바이스의 데이터 처리 능력으로서의 캐이퍼빌리티에 기초하여 구분한 서브 트리를 설정하고, 각각의 서브 트리의 관리 주체인 엔티티에 있어서, 엔티티 내에서 유효한 서브 유효화 키 블록(서브 EKB)을 생성함과 함께, 키 발행 센터(KDC)에서 엔티티의 캐이퍼빌리티 정보를 관리하여, 공통의 캐이퍼빌리티를 갖는 엔티티에 있어서만 복호 가능한 유효화 키 블록(EKB)을 생성하는 구성으로 했기 때문에, 특정한 디바이스에 있어서만 처리 가능한 데이터를 그 디바이스에 있어서만 복호 가능한 데이터로서 제공할 수 있다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 시스템 및 방법에 따르면, 키 발행 센터(KDC)는 복수의 엔티티 각각의 식별자, 캐이퍼빌리티 정보, 서브 유효화 키 블록(서브 EKB) 정보를 대응시킨 캐이퍼빌리티 관리 테이블에 기초하여 디바이스에 대한 배신 데이터의 처리 가능한 엔티티를 선택하여, 여러가지 캐이퍼빌리티에 따른 여러가지 EKB를 생성할 수 있다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 시스템 및 방법에 따르면, 복수의 디바이스를 리프로 구성한 트리의 루트에서 리프까지의 패스 상의 루트, 노드, 및 리프에 각각 키를 대응시킨 키 트리를 구성하는 부분 트리로서의 서브 트리를 관리하고, 서브 트리에 속하는 노드 또는 리프에 대응하여 설정되는 키만이 기초한 서브 유효화 키 블록(서브 EKB)을 생성하는 복수의 엔티티를 설정하고, 복수의 엔티티가 생성하는 서브 유효화 키 블록(서브 EKB)을 이용하여, 선택된 엔티티에 있어서만 복호 가능한 유효화 키 블록(EKB)을 생성하여 배신하는 구성으로 했기 때문에, 계층 구조의 키 트리 구성을 분해하여 관리할 수 있고, 디바이스에 따른 미세한 처리를 행할 수 있다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 시스템 및 방법에 따르면, 엔티티에서의 디바이스 또는 엔티티의 리보크 처리가 실행 가능하고, 일괄적인 디바이스 관리인 경우의 디바이스 증대에 따른 처리량의 증가가 방지된다.

또한, 본 발명에 따른 암호 키 블록을 이용한 정보 처리 시스템 및 방법에 따르면, 각 엔티티의 말단 노드에 리저브 노드를 설정하는 구성으로 했기 때문에, 관리 디바이스 또는 관리 엔티티의 증가에도 대응할 수 있다.

(57) 청구의 범위

청구항 1.

복수의 디바이스를 리프로 구성한 트리의 루트에서 리프까지의 패스 상의 루트, 노드, 및 리프에 각각 키를 대응시킨 키 트리를 구성하고, 상기 키 트리를 구성하는 패스를 선택하여 선택 패스 상의 키 갱신, 및 하위 키에 의한 상위 키의 암호화 처리를 실행하여 특정 디바이스에 있어서만 복호 가능한 유효화 키 블록(EKB)을 생성하여 디바이스에 제공하는 암호 키 블록을 이용한 정보 처리 시스템에 있어서,

상기 키 트리의 일부를 구성하고, 디바이스의 데이터 처리 능력으로서의 캐이퍼빌리티에 기초하여 구분된 서브 트리를 관리하고, 상기 서브 트리에 속하는 노드 또는 리프에 대응하여 설정되는 키에만 기초한 서브 유효화 키 블록(서브 EKB)을 생성하는 복수의 엔티티와,

상기 복수의 엔티티의 캐이퍼빌리티 정보를 관리하고, 공통의 캐이퍼빌리티를 갖는 엔티티가 생성하는 서브 유효화 키 블록(서브 EKB)을 이용하여, 공통의 캐이퍼빌리티를 갖는 엔티티에 있어서만 복호 가능한 유효화 키 블록(EKB)을 생성하는 키 발행 센터(KDC)

를 갖는 암호 키 블록을 이용한 정보 처리 시스템.

청구항 2.

제1항에 있어서,

상기 키 발행 센터(KDC)는, 복수의 엔티티 각각의 식별자와, 엔티티 각각의 캐이퍼빌리티 정보와, 엔티티 각각의 서브 유효화 키 블록(서브 EKB) 정보를 대응시킨 캐이퍼빌리티 관리 테이블을 갖고, 상기 캐이퍼빌리티 관리 테이블에 기초하여 디바이스에 대한 배신 데이터의 처리 가능한 엔티티를 선택하여, 상기 선택 엔티티 산하의 디바이스에서만 복호 가능한 유효화 키 블록(EKB)을 생성하는 구성을 갖는 것을 특징으로 하는 암호 키 블록을 이용한 정보 처리 시스템.

청구항 3.
제1항에 있어서,

상기 키 트리에 대한 신규 추가 엔티티는, 상기 신규 엔티티 내의 서브 트리 내의 노드 또는 리프에 대응하여 설정되는 키에만 기초한 서브 유효화 키 블록(서브 EKB)을 생성하고, 상기 키 발행 센터(KDC)에 대한 서브 EKB의 등록 처리를 실행함과 함께, 자신의 엔티티의 캐이퍼빌리티 정보의 통지 처리를 실행하는 구성인 것을 특징으로 하는 암호 키 블록을 이용한 정보 처리 시스템.

청구항 4.
제1항에 있어서,

상기 복수의 엔티티는, 하나의 엔티티의 최하단의 말단 노드를 다른 엔티티의 정점 노드(서브 루트)로서 구성한 상위 엔티티 및 하위 엔티티의 계층화 구조를 갖는 것을 특징으로 하는 암호 키 블록을 이용한 정보 처리 시스템.

청구항 5.
제1항에 있어서,

상기 복수의 엔티티의 각각은, 자신의 엔티티에 속하는 서브 트리를 구성하는 노드 또는 리프에 대응하는 키의 설정, 갱신 처리 권한을 갖는 구성인 것을 특징으로 하는 암호 키 블록을 이용한 정보 처리 시스템.

청구항 6.
제1항에 있어서,

상기 복수의 엔티티 중, 엔티티 내의 최하단 리프를 개개의 디바이스에 대응하는 리프로 한 최하층의 엔티티에 속하는 디바이스의 각각은, 자신이 속하는 엔티티의 정점 노드(서브 루트)로부터 자신의 디바이스에 대응하는 리프에 이르는 패스 상의 노드, 리프에 설정된 노드 키 및 리프 키를 저장한 구성을 갖는 것을 특징으로 하는 암호 키 블록을 이용한 정보 처리 시스템.

청구항 7.
제1항에 있어서,

상기 복수의 엔티티의 각각은, 자신의 엔티티의 하위에, 또한 자기 관리 엔티티를 추가하기 위해서 자신의 엔티티 내의 최하단의 노드 또는 리프 중의 1이상의 노드 또는 리프를 리저브 노드로서 보류하여 설정한 구성을 갖는 것을 특징으로 하는 암호 키 블록을 이용한 정보 처리 시스템.

청구항 8.
제1항에 있어서,

신규 엔티티를 말단 노드에 추가하는 상위 엔티티는 신규 엔티티의 서브 트리를 설정하는 노드인 상위 엔티티 말단 노드에 대응하는 키를 상기 신규 엔티티의 정점 노드(서브 루트) 키로서 설정하는 구성인 것을 특징으로 하는 암호 키 블록을 이용한 정보 처리 시스템.

청구항 9.
제1항에 있어서,

디바이스의 리보크 처리를 실행하는 엔티티는, 엔티티 내의 정점 노드(서브 루트)로부터 리보크 디바이스에 대응하는 리프에 이르는 패스 상의 노드에 설정된 노드 키를 갱신하고, 갱신 노드 키를 리보크 디바이스 이외의 리프 디바이스에 있어서만 복호 가능한 암호화 키로서 구성한 갱신 서브 EKB를 생성하여 상위 엔티티에 송신하고, 상위 엔티티는 갱신 서브 EKB를 제공한 말단 노드로부터 자신의 서브 루트에 이르는 패스 상의 노드 키를 갱신한 갱신 서브 EKB를 생성하여 다시 상위 엔티티에 송신하고, 최상위 엔티티까지, 엔티티 단위의 갱신 서브 EKB 생성 및 송신 처리를 순차적으로 실행하여 리보크 디바이스로부터 루트에 이르는 패스 상의 노드 키 갱신을 행하고, 키 갱신에 의해 생성된 갱신 서브 EKB의 상기 키 발행 센터(KDC)에의 등록 처리를 행함으로써, 디바이스의 리보크 처리를 실행하는 구성을 갖는 것을 특징으로 하는 암호 키 블록을 이용한 정보 처리 시스템.

청구항 10.
제1항에 있어서,

하위 엔티티의 리보크 처리를 실행하는 엔티티는, 엔티티 내의 정점 노드(서브 루트)로부터 리보크 엔티티에 대응하는 말단 노드에 이르는 패스 상의 노드에 설정된 노드 키를 갱신한 갱신 서브 EKB를 생성하여 상위 엔티티에 송신하고, 상위 엔티티는 갱신 서브 EKB를 제공한 말단 노드로부터 자신의 서브 루트에 이르는 패스 상의 노드 키를 갱신한 갱신 서브 EKB를 생성하여 다시 상위 엔티티에 송신하고, 최상위 엔티티까지 엔티티 단위의 갱신 서브 EKB 생성 및 송신 처리를 순차적으로 실행하여 리보크 엔티티로부터 루트에 이르는 패스 상의 노드 키 갱신을 행하고, 키 갱신에 의해 생성된 갱신 서브 EKB의 상기 키 발행 센터(KDC)에의 등록 처리를 행함으로써, 엔티티 단위의 리보크 처리를 실행하는 구성을 갖는 것을 특징으로 하는 암호 키 블록을 이용한 정보 처리 시스템.

청구항 11.
제1항에 있어서,

하위 엔티티의 리보크 처리를 실행하는 엔티티는, 엔티티 내의 정점 노드(서브 루트)로부터 리보크·엔티티에 대응하는 말단 노드에 이르는 패스 상의, 상기 말단 노드를 제외한 노드에 설정된 노드 키를 갱신한 갱신 서브 EKB를 생성하여 상위 엔티티에 송신하고, 상위 엔티티는 갱신 서브 EKB를 제공한 말단 노드로부터 자신의 서브 루트에 이르는 패스 상의 노드 키를 갱신한 갱신 서브 EKB를 생성하여 다시 상위 엔티티에 송신하고, 최상위 엔티티까지, 엔티티 단위의 갱신 서브 EKB 생성 및 송신 처리를 순차적으로 실행하여 리보크·엔티티로부터 루트에 이르는 패스 상의 리보크·엔티티에 대응하는 말단 노드를 제외한 노드 키 갱신을 행하고, 키 갱신에 의해 생성된 갱신 서브 EKB의 상기 키 발행 센터(KDC)에의 등록 처리를 행함으로써, 엔티티 단위의 리보크 처리를 실행하는 구성을 갖는 것을 특징으로 하는 암호 키 블록을 이용한 정보 처리 시스템.

청구항 12.
복수의 디바이스를 리프로 구성한 트리의 루트에서 리프까지의 패스 상의 루트, 노드, 및 리프에 각각 키를 대응시킨 키 트리를 구성하고, 상기 키 트리를 구성하는 패스를 선택하여 선택 패스 상의 키 갱신, 및 하위 키에 의한 상위 키의 암호화 처리를 실행하여 특정 디바이스에 있어서만 복호 가능한 유효화 키 블록(EKB)을 생성하여 디바이스에 제공하는 정보 처리 시스템에 있어서의 암호 키 블록을 이용한 정보 처리 방법에 있어서,

상기 키 트리의 일부를 구성하고, 디바이스의 데이터 처리 능력으로서의 캐이퍼빌리티에 기초하여 구분된 서브 트리를 관리하는 엔티티에 있어서, 각 엔티티의 서브 트리에 속하는 노드 또는 리프에 대응하여 설정되는 키에만 기초한 서브 유효화 키 블록(서브 EKB)을 생성하는 단계와,

상기 복수의 엔티티의 캐이퍼빌리티 정보를 보유하는 키 발행 센터(KDC)에서 상기 복수의 엔티티의 캐이퍼빌리티 정보에 기초하여 공통의 캐이퍼빌리티를 갖는 엔티티가 생성하는 서브 유효화 키 블록(서브 EKB)을 추출하여 공통의 캐이퍼빌리티를 갖는 엔티티에 있어서만 복호 가능한 유효화 키 블록(EKB)을 생성하는 단계

를 포함하는 암호 키 블록을 이용한 정보 처리 방법.

청구항 13.
제12항에 있어서,

상기 키 발행 센터(KDC)에 있어서의 유효화 키 블록(EKB) 생성 단계는,

공통의 캐이퍼빌리티를 갖는 엔티티를 선택하는 엔티티 선택 단계와,

상기 엔티티 선택 단계에서 선택된 엔티티에 의해 구성되는 엔티티·트리를 생성하는 단계와,

상기 엔티티·트리를 구성하는 노드 키를 갱신하는 노드 키 갱신 단계와,

상기 노드 키 갱신 단계에서 갱신한 노드 키, 및 선택 엔티티의 서브 EKB에 기초하여 선택 엔티티에 있어서만 복호 가능한 유효화 키 블록(EKB)을 생성하는 단계

를 포함하는 것을 특징으로 하는 암호 키 블록을 이용한 정보 처리 방법.

청구항 14.
제12항에 있어서,

상기 키 발행 센터(KDC)는 복수의 엔티티 각각의 식별자와, 엔티티 각각의 캐이퍼빌리티 정보와, 엔티티 각각의 서브 유효화 키 블록(서브 EKB) 정보를 대응시킨 캐이퍼빌리티 관리 테이블을 갖고, 상기 캐이퍼빌리티 관리 테이블에 기초하여 디바이스에 대한 배신 데이터의 처리 가능한 엔티티를 선택하여, 상기 선택 엔티티 산하의 디바이스에서만 복호 가능한 유효화 키 블록(EKB)을 생성하는 것을 특징으로 하는 암호 키 블록을 이용한 정보 처리 방법.

청구항 15.
제12항에 있어서,

상기 키 트리에 대한 신규 추가 엔티티는, 상기 신규 엔티티 중의 서브 트리 내의 노드 또는 리프에 대응하여 설정되는 키에만 기초한 서브 유효화 키 블록(서브 EKB)을 생성하고, 상기 키 발행 센터(KDC)에 대한 서브 EKB의 등록 처리를 실행함과 함께, 자신의 엔티티의 캐이퍼빌리티 정보의 통지 처리를 실행하는 것을 특징으로 하는 암호 키 블록을 이용한 정보 처리 방법.

청구항 16.
제12항에 있어서,

상기 복수의 엔티티의 각각은, 자신의 엔티티에 속하는 서브 트리를 구성하는 노드 또는 리프에 대응하는 키의 설정, 갱신 처리를 실행하는 것을 특징으로 하는 암호 키 블록을 이용한 정보 처리 방법.

청구항 17.
제12항에 있어서,

신규 엔티티를 말단 노드에 추가하는 상위 엔티티는, 신규 엔티티의 서브 트리를 설정하는 노드인 상위 엔티티 말단 노드에 대응하는 키를 상기 신규 엔티티의 정점 노드(서브 루트) 키로서 설정하는 것을 특징으로 하는 암호 키 블록을 이용한 정보 처리 방법.

청구항 18.
제12항에 있어서,

디바이스의 리보크 처리를 실행하는 엔티티는, 엔티티 내의 정점 노드(서브 루트)로부터 리보크 디바이스에 대응하는 리프에 이르는 패스 상의 노드에 설정된 노드 키를 갱신하고, 갱신 노드 키를 리보크 디바이스 이외의 리프 디바이스에 있어서만 복호 가능한 암호화 키로서 구성한 갱신 서브 EKB를 생성하여 상위 엔티티에 송신하고, 상위 엔티티는 갱신 서브 EKB를 제공한 말단 노드로부터 자신의 서브 루트에 이르는 패스 상의 노드 키를 갱신한 갱신 서브 EKB를 생성하여 다시 상위 엔티티에 송신하고, 최상위 엔티티까지, 엔티티 단위의 갱신 서브 EKB 생성 및 송신 처리를 순차적으로 실행하여 리보크 디바이스로부터 루트에 이르는 패스 상의 노드 키 갱신을 행하고, 키 갱신에 의해 생성된 갱신 서브 EKB의 상기 키 발행 센터(KDC)에의 등록 처리를 행함으로써, 디바이스의 리보크 처리를 실행하는 것을 특징으로 하는 암호 키 블록을 이용한 정보 처리 방법.

청구항 19.
제12항에 있어서,

하위 엔티티의 리보크 처리를 실행하는 엔티티는, 엔티티 내의 정점 노드(서브 루트)로부터 리보크·엔티티에 대응하는 말단 노드에 이르는 패스 상의 노드에 설정된 노드 키를 갱신한 갱신 서브 EKB를 생성하여 상위 엔티티에 송신하고, 상위 엔티티는 갱신 서브 EKB를 제공한 말단 노드부터 자신의 서브 루트에 이르는 패스 상의 노드 키를 갱신한 갱신 서브 EKB를 생성하여 다시 상위 엔티티에 송신하고, 최상위 엔티티까지, 엔티티 단위의 갱신 서브 EKB 생성 및 송신 처리를 순차적으로 실행하여 리보크·엔티티로부터 루트에 이르는 패스 상의 노드 키 갱신을 행하고, 키 갱신에 의해 생성된 갱신 서브 EKB의 상기 키 발행 센터(KDC)에의 등록 처리를 행함으로써, 엔티티 단위의 리보크 처리를 실행하는 것을 특징으로 하는 암호 키 블록을 이용한 정보 처리 방법.

청구항 20.
제12항에 있어서,

하위 엔티티의 리보크 처리를 실행하는 엔티티는, 엔티티 내의 정점 노드(서브 루트)로부터 리보크·엔티티에 대응하는 말단 노드에 이르는 패스 상의, 상기 말단 노드를 제외한 노드에 설정된 노드 키를 갱신한 갱신 서브 EKB를 생성하여 상위 엔티티에 송신하고, 상위 엔티티는 갱신 서브 EKB를 제공한 말단 노드로부터 자신의 서브 루트에 이르는 패스 상의 노드 키를 갱신한 갱신 서브 EKB를 생성하여 다시 상위 엔티티에 송신하고, 최상위 엔티티까지, 엔티티 단위의 갱신 서브 EKB 생성 및 송신 처리를 순차적으로 실행하여 리보크·엔티티로부터 루트에 이르는 패스 상의 리보크·엔티티에 대응하는 말단 노드를 제외한 노드 키 갱신을 행하고, 키 갱신에 의해 생성된 갱신 서브 EKB의 상기 키 발행 센터(KDC)에의 등록 처리를 행함으로써, 엔티티 단위의 리보크 처리를 실행하는 것을 특징으로 하는 암호 키 블록을 이용한 정보 처리 방법.

청구항 21.
복수의 디바이스를 리프로 구성한 트리의 루트에서 리프까지의 패스 상의 루트, 노드, 및 리프에 각각 키를 대응시킨 키 트리를 구성하고, 상기 키 트리를 구성하는 패스를 선택하여 선택 패스 상의 키 갱신, 및 하위 키에 의한 상위 키의 암호화 처리를 실행하여 특정 디바이스에 있어서만 복호 가능한 유효화 키 블록(EKB)을 생성하여 디바이스에 제공하는 정보 처리 시스템에 있어서의 유효화 키 블록(EKB) 생성 처리를 컴퓨터 시스템 상에서 실행시키는 컴퓨터 프로그램을 제공하는 프로그램 제공 매체에 있어서, 상기 컴퓨터 프로그램은,

상기 키 트리의 일부를 구성하고, 디바이스의 데이터 처리 능력으로서의 케이퍼빌리티에 기초하여 구분된 서브 트리를 관리하는 엔티티에 있어서, 각 엔티티의 서브 트리에 속하는 노드 또는 리프에 대응하여 설정되는 키에만 기초한 서브 유효화 키 블록(서브 EKB)을 생성하는 단계와,

상기 복수의 엔티티의 케이퍼빌리티 정보를 보유하는 키 발행 센터(KDC)에서 상기 복수의 엔티티의 케이퍼빌리티 정보에 기초하여 공통의 케이퍼빌리티를 갖는 엔티티가 생성하는 서브 유효화 키 블록(서브 EKB)을 추출하여 공통의 케이퍼빌리티를 갖는 엔티티에 있어서만 복호 가능한 유효화 키 블록(EKB)을 생성하는 단계

를 포함하는 것을 특징으로 하는 프로그램 제공 매체.

청구항 22.
복수의 디바이스를 리프로 구성한 트리의 루트에서 리프까지의 패스 상의 루트, 노드, 및 리프에 각각 키를 대응시킨 키 트리를 구성하고, 상기 키 트리를 구성하는 패스를 선택하여 선택 패스 상의 키 갱신, 및 하위 키에 의한 상위 키의 암호화 처리를 실행하여 특정 디바이스에 있어서만 복호 가능한 유효화 키 블록(EKB)을 생성하여 디바이스에 제공하는 암호 키 블록을 이용한 정보 처리 시스템에 있어서,

상기 키 트리를 구성하는 부분 트리로서의 서브 트리를 관리하고, 상기 서브 트리에 속하는 노드 또는 리프에 대응하여 설정되는 키에만 기초한 서브 유효화 키 블록(서브 EKB)을 생성하는 복수의 엔티티와,

상기 복수의 엔티티가 생성하는 서브 유효화 키 블록(서브 EKB)을 이용하여, 선택된 엔티티에 있어서만 복호 가능한 유효화 키 블록(EKB)을 생성하는 키 발행 센터(KDC)

를 갖는 암호 키 블록을 이용한 정보 처리 시스템.

청구항 23.
제22항에 있어서,

상기 복수의 엔티티는 하나의 엔티티의 최하단의 말단 노드를 다른 엔티티의 정점 노드(서브 루트)로서 구성한 상위 엔티티 및 하위 엔티티의 계층화 구조를 갖는 것을 특징으로 하는 암호 키 블록을 이용한 정보 처리 시스템.

청구항 24.
제22항에 있어서,

상기 복수의 엔티티의 각각은 자신의 엔티티에 속하는 서브 트리를 구성하는 노드 또는 리프에 대응하는 키의 설정, 갱신 처리 권한을 갖는 구성인 것을 특징으로 하는 암호 키 블록을 이용한 정보 처리 시스템.

청구항 25.
제22항에 있어서,

상기 복수의 엔티티 중, 엔티티 내의 최하단 리프를 개개의 디바이스에 대응하는 리프로 한 최하층의 엔티티에 속하는 디바이스의 각각은 자신이 속하는 엔티티의 정점 노드(서브 루트)로부터 자신의 디바이스에 대응하는 리프에 이르는 패스 상의 노드, 리프에 설정된 노드 키 및 리프 키를 저장한 구성을 갖는 것을 특징으로 하는 암호 키 블록을 이용한 정보 처리 시스템.

청구항 26.
제22항에 있어서,

상기 복수의 엔티티의 각각은 자신의 엔티티의 하위에, 자기 관리 엔티티를 더 추가하기 위해서, 자신의 엔티티 내의 최하단의 노드 또는 리프 중의 1 이상의 노드 또는 리프를 리저브 노드로서 보류하여 설정한 구성을 갖는 것을 특징으로 하는 암호 키 블록을 이용한 정보 처리 시스템.

청구항 27.
제22항에 있어서,

신규 엔티티를 말단 노드에 추가하는 상위 엔티티는, 신규 엔티티의 서브 트리를 설정하는 노드인 상위 엔티티 말단 노드에 대응하는 키를 상기 신규 엔티티의 정점 노드(서브 루트) 키로서 설정하는 구성인 것을 특징으로 하는 암호 키 블록을 이용한 정보 처리 시스템.

청구항 28.
제22항에 있어서,

신규 추가 엔티티는, 상기 신규 엔티티 내의 서브 트리 내의 노드 또는 리프에 대응하여 설정되는 키에만 기초한 서브 유효화 키 블록(서브 EKB)을 생성하고, 상기 키 발행 센터(KDC)에 대한 서브 EKB의 등록 처리를 실행하는 구성인 것을 특징으로 하는 암호 키 블록을 이용한 정보 처리 시스템.

청구항 29.
제22항에 있어서,

디바이스의 리보크 처리를 실행하는 엔티티는, 엔티티 내의 정점 노드(서브 루트)로부터 리보크 디바이스에 대응하는 리프에 이르는 패스 상의 노드에 설정된 노드 키를 갱신하고, 갱신 노드 키를 리보크 디바이스 이외의 리프 디바이스에 있어서만 복호 가능한 암호화 키로서 구성한 갱신 서브 EKB를 생성하여 상위 엔티티에 송신하고, 상위 엔티티는 갱신 서브 EKB를 제공한 말단 노드로부터 자신의 서브 루트에 이르는 패스 상의 노드 키를 갱신한 갱신 서브 EKB를 생성하여 다시 상위 엔티티에 송신하고, 최상위 엔티티까지, 엔티티 단위의 갱신 서브 EKB 생성 및 송신 처리를 순차적으로 실행하여 리보크 디바이스로부터 루트에 이르는 패스 상의 노드 키 갱신을 행하고, 키 갱신에 의해 생성된 갱신 서브 EKB의 상기 키 발행 센터(KDC)에의 등록 처리를 행함으로써, 디바이스의 리보크 처리를 실행하는 구성을 갖는 것을 특징으로 하는 암호 키 블록을 이용한 정보 처리 시스템.

청구항 30.
제22항에 있어서,

하위 엔티티의 리보크 처리를 실행하는 엔티티는, 엔티티 내의 정점 노드(서브 루트)로부터 리보크·엔티티에 대응하는 말단 노드에 이르는 패스 상의 노드에 설정된 노드 키를 갱신한 갱신 서브 EKB를 생성하여 상위 엔티티에 송신하고, 상위 엔티티는 갱신 서브 EKB를 제공한 말단 노드로부터 자신의 서브 루트에 이르는 패스 상의 노드 키를 갱신한 갱신 서브 EKB를 생성하여 다시 상위 엔티티에 송신하고, 최상위 엔티티까지, 엔티티 단위의 갱신 서브 EKB 생성 및 송신 처리를 순차적으로 실행하여 리보크·엔티티로부터 루트에 이르는 패스 상의 노드 키 갱신을 행하고, 키 갱신에 의해 생성된 갱신 서브 EKB의 상기 키 발행 센터(KDC)에의 등록 처리를 행함으로써, 엔티티 단위의 리보크 처리를 실행하는 구성을 갖는 것을 특징으로 하는 암호 키 블록을 이용한 정보 처리 시스템.

청구항 31.
제22항에 있어서,

하위 엔티티의 리보크 처리를 실행하는 엔티티는, 엔티티 내의 정점 노드(서브 루트)로부터 리보크·엔티티에 대응하는 말단 노드에 이르는 패스 상의, 상기 말단 노드를 제외한 노드에 설정된 노드 키를 갱신한 갱신 서브 EKB를 생성하여 상위 엔티티에 송신하고, 상위 엔티티는 갱신 서브 EKB를 제공한 말단 노드로부터 자신의 서브 루트에 이르는 패스 상의 노드 키를 갱신한 갱신 서브 EKB를 생성하여 다시 상위 엔티티에 송신하고, 최상위 엔티티까지, 엔티티 단위의 갱신 서브 EKB 생성 및 송신 처리를 순차적으로 실행하여 리보크·엔티티로부터 루트에 이르는 패스 상의 리보크·엔티티에 대응하는 말단 노드를 제외한 노드 키 갱신을 행하고, 키 갱신에 의해 생성된 갱신 서브 EKB의 상기 키 발행 센터(KDC)에의 등록 처리를 행함으로써, 엔티티 단위의 리보크 처리를 실행하는 구성을 갖는 것을 특징으로 하는 암호 키 블록을 이용한 정보 처리 시스템.

청구항 32.
제22항에 있어서,

상기 엔티티는 디바이스 종류, 서비스 종류, 관리 수단 종류 등의 공통의 카테고리에 속하는 디바이스 또는 엔티티의 관리 주체로서 구성되는 것을 특징으로 하는 암호 키 블록을 이용한 정보 처리 시스템.

청구항 33.
복수의 디바이스를 리프로 구성한 트리의 루트에서 리프까지의 패스 상의 루트, 노드 및 리프에 각각 키를 대응시킨 키 트리를 구성하고, 상기 키 트리를 구성하는 패스를 선택하여 선택 패스 상의 키 갱신, 및 하위 키에 의한 상위 키의 암호화 처리를 실행하여 특정 디바이스에 있어서만 복호 가능한 유효화 키 블록(EKB)을 생성하여 디바이스에 제공하는 정보 처리 시스템에 있어서의 암호 키 블록을 이용한 정보 처리 방법에 있어서,

상기 키 트리를 구성하는 부분 트리로서의 서브 트리를 관리하는 복수의 엔티티에 있어서, 상기 서브 트리에 속하는 노드 또는 리프에 대응하여 설정되는 키에만 기초한 서브 유효화 키 블록(서브 EKB)을 생성하는 단계와,

키 발행 센터(KDC)에서 상기 복수의 엔티티가 생성하는 서브 유효화 키 블록(서브 EKB)을 이용하여, 선택된 엔티티에 있어서만 복호 가능한 유효화 키 블록(EKB)을 생성하는 단계

를 포함하는 암호 키 블록을 이용한 정보 처리 방법.

청구항 34.
제33항에 있어서,

상기 복수의 엔티티의 각각은, 자신의 엔티티에 속하는 서브 트리를 구성하는 노드 또는 리프에 대응하는 키의 설정, 갱신 처리를 실행하는 것을 특징으로 하는 암호 키 블록을 이용한 정보 처리 방법.

청구항 35.
제33항에 있어서,

신규 엔티티를 말단 노드에 추가하는 상위 엔티티는, 신규 엔티티의 서브 트리를 설정하는 노드인 상위 엔티티 말단 노드에 대응하는 키를 상기 신규 엔티티의 정점 노드(서브 루트) 키로서 설정하는 것을 특징으로 하는 암호 키 블록을 이용한 정보 처리 방법.

청구항 36.
제33항에 있어서,

신규 추가 엔티티는, 상기 신규 엔티티 내의 서브 트리 내의 노드 또는 리프에 대응하여 설정되는 키에만 기초한 서브 유효화 키 블록(서브 EKB)을 생성하고, 상기 키 발행 센터(KDC)에 대한 서브 EKB의 등록 처리를 실행하는 것을 특징으로 하는 암호 키 블록을 이용한 정보 처리 방법.

청구항 37.
제33항에 있어서,

디바이스의 리보크 처리를 실행하는 엔티티는, 엔티티 내의 정점 노드(서브 루트)로부터 리보크 디바이스에 대응하는 리프에 이르는 패스 상의 노드에 설정된 노드 키를 갱신하고, 갱신 노드 키를 리보크 디바이스 이외의 리프 디바이스에 있어서만 복호 가능한 암호화 키로서 구성한 갱신 서브 EKB를 생성하여 상위 엔티티에 송신하고, 상위 엔티티는 갱신 서브 EKB를 제공한 말단 노드로부터 자신의 서브 루트에 이르는 패스 상의 노드 키를 갱신한 갱신 서브 EKB를 생성하여 다시 상위 엔티티에 송신하고, 최상위 엔티티까지, 엔티티 단위의 갱신 서브 EKB 생성 및 송신 처리를 순차적으로 실행하여 리보크 디바이스로부터 루트에 이르는 패스 상의 노드 키 갱신을 행하고, 키 갱신에 의해 생성된 갱신 서브 EKB의 상기 키 발행 센터(KDC)에의 등록 처리를 행함으로써, 디바이스의 리보크 처리를 실행하는 것을 특징으로 하는 암호 키 블록을 이용한 정보 처리 방법.

청구항 38.
제33항에 있어서,

하위 엔티티의 리보크 처리를 실행하는 엔티티는 엔티티 내의 정점 노드(서브 루트)로부터 리보크·엔티티에 대응하는 말단 노드에 이르는 패스 상의 노드에 설정된 노드 키를 갱신한 갱신 서브 EKB를 생성하여 상위 엔티티에 송신하고, 상위 엔티티는 갱신 서브 EKB를 제공한 말단 노드로부터 자신의 서브 루트에 이르는 패스 상의 노드 키를 갱신한 갱신 서브 EKB를 생성하여 다시 상위 엔티티에 송신하고, 최상위 엔티티까지, 엔티티 단위의 갱신 서브 EKB 생성 및 송신 처리를 순차적으로 실행하여 리보크·엔티티로부터 루트에 이르는 패스 상의 노드 키 갱신을 행하고, 키 갱신에 의해 생성된 갱신 서브 EKB의 상기 키 발행 센터(KDC)에의 등록 처리를 행함으로써, 엔티티 단위의 리보크 처리를 실행하는 것을 특징으로 하는 암호 키 블록을 이용한 정보 처리 방법.

청구항 39.
제33항에 있어서,

하위 엔티티의 리보크 처리를 실행하는 엔티티는 엔티티 내의 정점 노드(서브 루트)로부터 리보크·엔티티에 대응하는 말단 노드에 이르는 패스 상의, 상기 말단 노드를 제외한 노드에 설정된 노드 키를 갱신한 갱신 서브 EKB를 생성하여 상위 엔티티에 송신하고, 상위 엔티티는 갱신 서브 EKB를 제공한 말단 노드로부터 자신의 서브 루트에 이르는 패스 상의 노드 키를 갱신한 갱신 서브 EKB를 생성하여 다시 상위 엔티티에 송신하고, 최상위 엔티티까지, 엔티티 단위의 갱신 서브 EKB 생성 및 송신 처리를 순차적으로 실행하여 리보크·엔티티로부터 루트에 이르는 패스 상의 리보크 엔티티에 대응하는 말단 노드를 제외한 노드 키 갱신을 행하고, 키 갱신에 의해 생성된 갱신 서브 EKB의 상기 키 발행 센터(KDC)에의 등록 처리를 행함으로써, 엔티티 단위의 리보크 처리를 실행하는 것을 특징으로 하는 암호 키 블록을 이용한 정보 처리 방법.

청구항 40.
복수의 디바이스를 리프로 구성한 트리의 루트에서 리프까지의 패스 상의 루트, 노드, 및 리프에 각각 키를 대응시킨 키 트리를 구성하고, 상기 키 트리를 구성하는 패스를 선택하여 선택 패스 상의 키 갱신, 및 하위 키에 의한 상위 키의 암호화 처리를 실행하여 특정 디바이스에 있어서만 복호 가능한 유효화 키 블록(EKB)을 생성하여 디바이스에 제공하는 정보 처리 시스템에 있어서의 유효화 키 블록(EKB) 생성 처리를 컴퓨터 시스템 상에서 실행시키는 컴퓨터 프로그램을 제공하는 프로그램 제공 매체에 있어서, 상기 컴퓨터 프로그램은

상기 키 트리를 구성하는 부분 트리로서의 서브 트리를 관리하는 복수의 엔티티에 있어서, 상기 서브 트리에 속하는 노드 또는 리프에 대응하여 설정되는 키에만 기초한 서브 유효화 키 블록(서브 EKB)을 생성하는 단계와,

키 발행 센터(KDC)에서 상기 복수의 엔티티가 생성하는 서브 유효화 키 블록(서브 EKB)을 이용하여, 선택된 엔티티에 있어서만 복호 가능한 유효화 키 블록(EKB)을 생성하는 단계

를 포함하는 것을 특징으로 하는 프로그램 제공 매체.

요약

복수의 디바이스를 리프로 한 트리의 루트, 노드, 리프에 키를 대응시킨 키 트리에, 디바이스의 데이터 처리 능력(캐이퍼빌리티)으로 구분한 서브 트리를 설정하고, 각 서브 트리의 관리 주체(엔티티)에 있어서, 엔티티 내에서 유효한 서브 유효화 키 블록을 생성하고, 키 발행 센터에서 엔티티의 캐이퍼빌리티 정보에 기초하여 공통 캐이퍼빌리티를 갖는 엔티티에 있어서만 복호 가능한 유효화 키 블록을 생성한다. 또한, 키 트리의 부분 트리(서브 트리)를 관리하여, 서브 트리에 속하는 노드 또는 리프에 대응하여 설정되는 키에만 기초한 서브 유효화 키 블록을 생성하고, 서브 유효화 키 블록을 이용하여, 선택된 엔티티에 있어서만 복호 가능한 유효화 키 블록을 생성한다. 이에 따라, 디바이스의 데이터 처리 능력에 따라서 유효화 키 블록을 생성하여 배신할 수 있고, 또한 계층 구조의 키 트리 구성을 분할하여 관리할 수 있다.

대표도

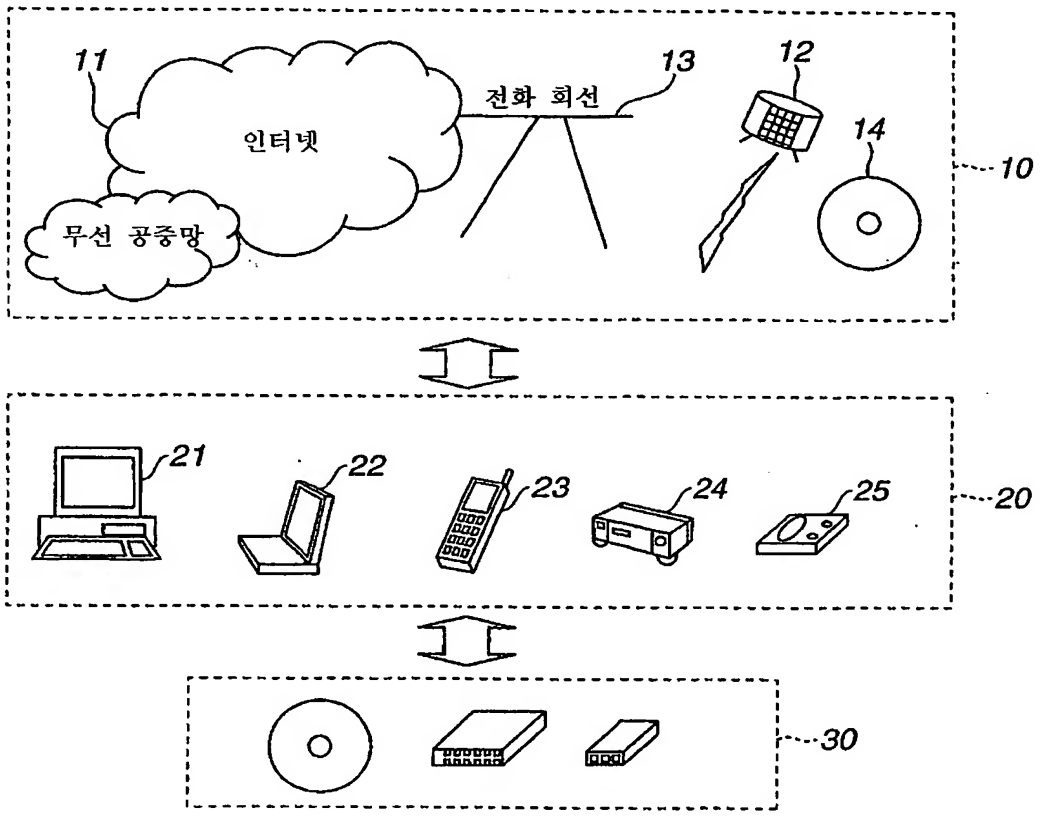
도40

색인어

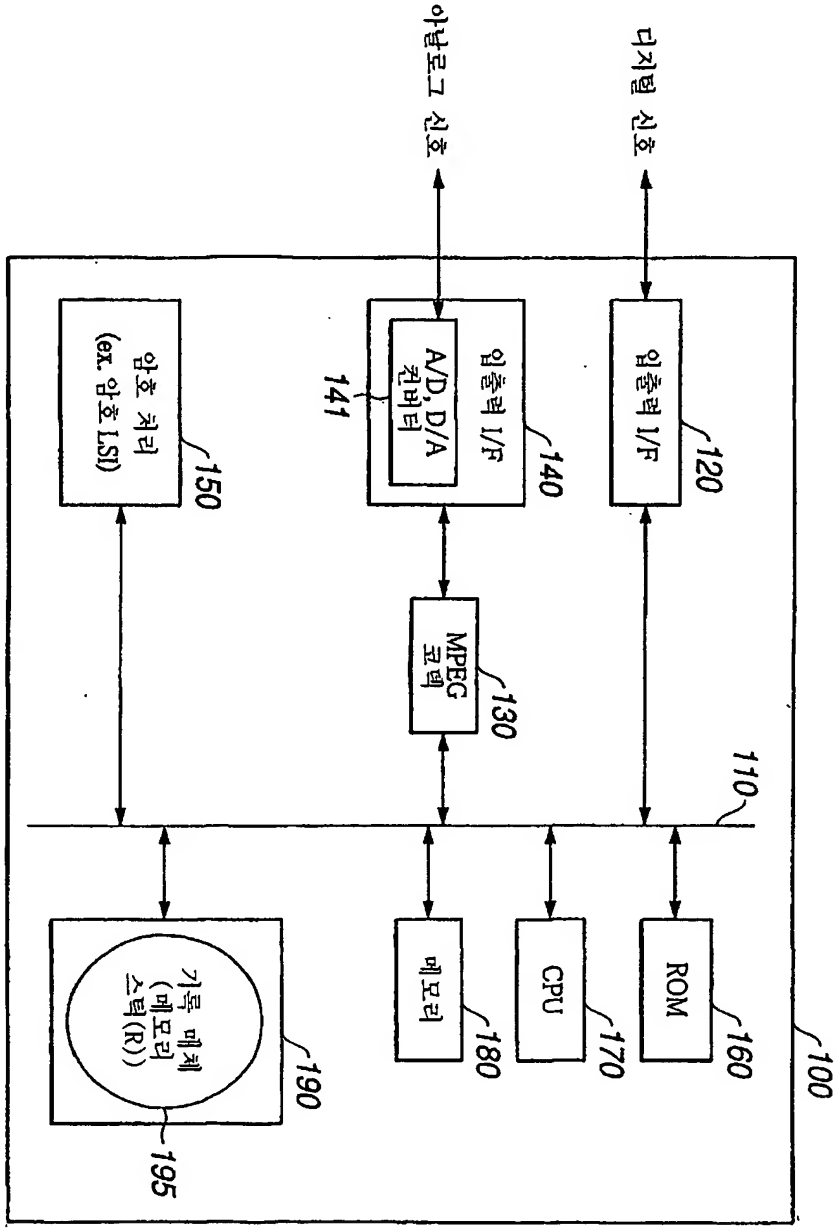
엔티티, 캐이퍼빌리티, 트리, 루프, 리프

도면

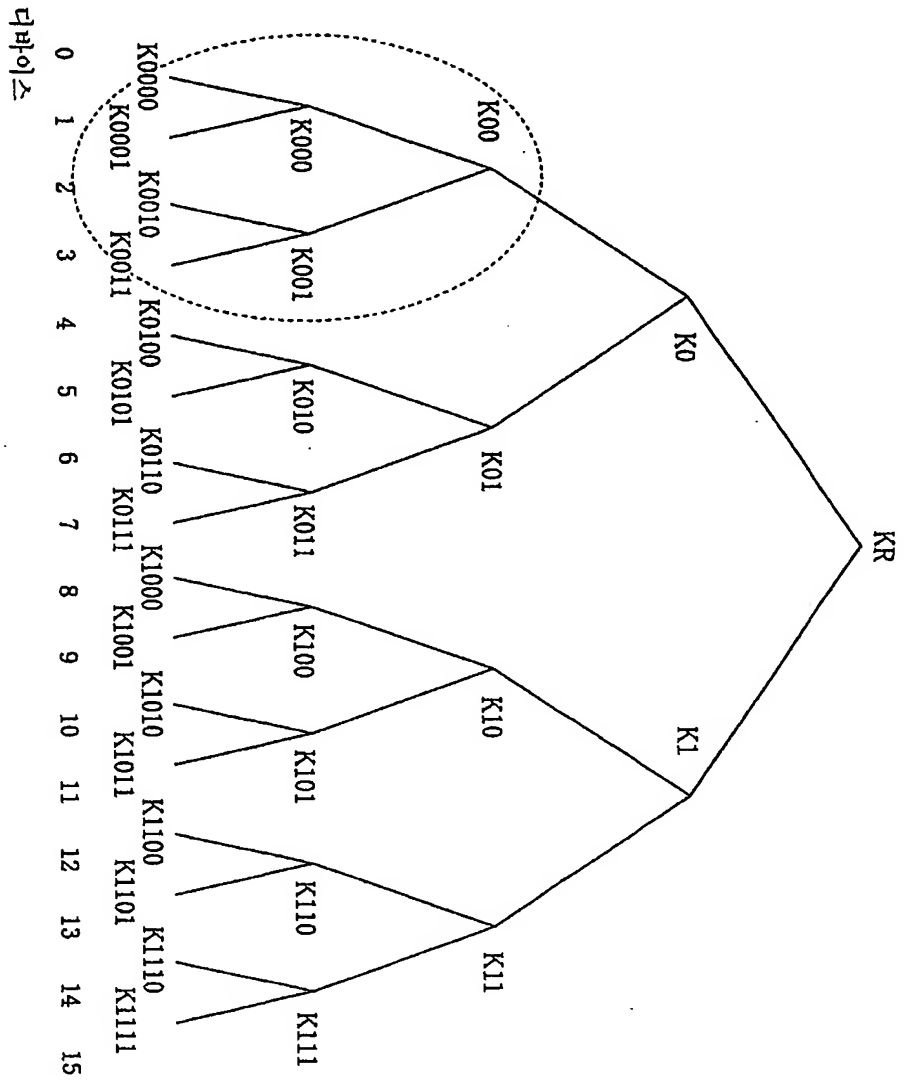
도면 1



도면 2


디지털 신호
아날로그 신호
입출력 I/F
120
입출력 I/F
140
A/D, D/A
컨버터
141
암호 처리
(ex. 암호 LSI)
150
MPEG
코덱
130
110
ROM
160
CPU
170
메모리
180
기록 매체
(메모리
스틱(R))
190
195
100

도면 3



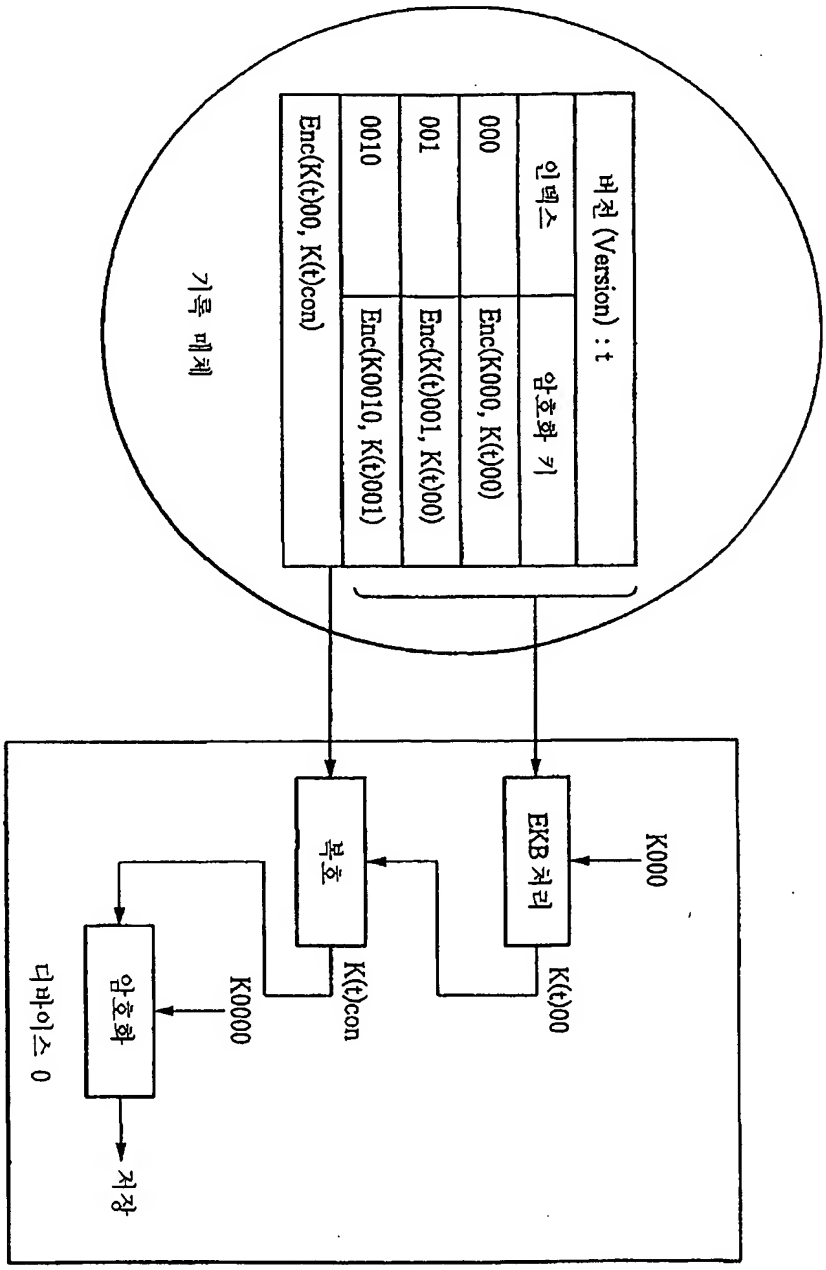
도면 4A

버전 (Version) : t	
인덱스	암호화 키
0	Enc(K(t)0, K(t)R)
00	Enc(K(t)00, K(t)0)
000	Enc(K000, K(t)00)
001	Enc(K(t)001, K(t)00)
0010	Enc(K0010, K(t)001)

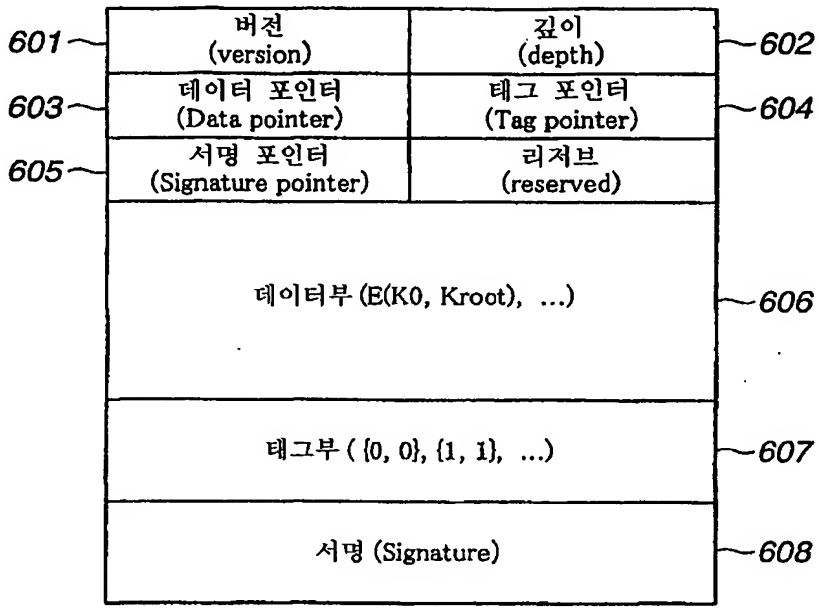
도면 4B

버전 (Version) : t	
인덱스	암호화 키
000	Enc(K000, K(t)00)
001	Enc(K(t)001, K(t)00)
0010	Enc(K0010, K(t)001)

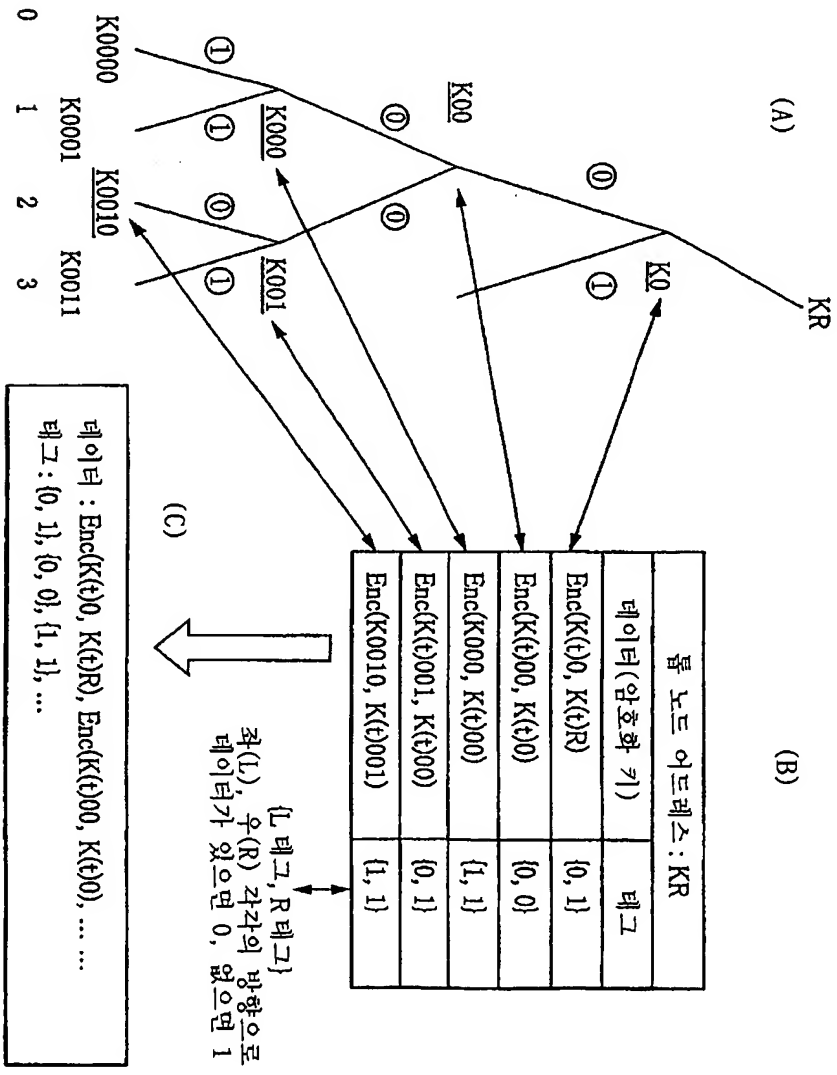
도면 5



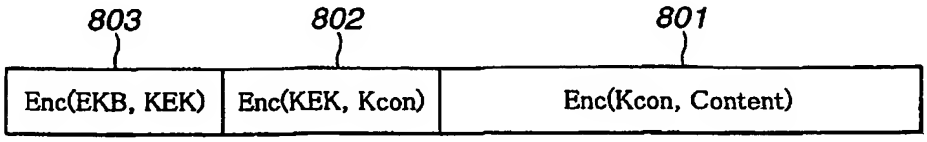
도면 6


601
버전 (version)
깊이 (depth)
602
603
데이터 포인터 (Data pointer)
태그 포인터 (Tag pointer)
604
605
서명 포인터 (Signature pointer)
리저브 (reserved)
데이터부 (E(K0, Kroot), ...)
606
태그부 ({0, 0}, {1, 1}, ...)
607
서명 (Signature)
608

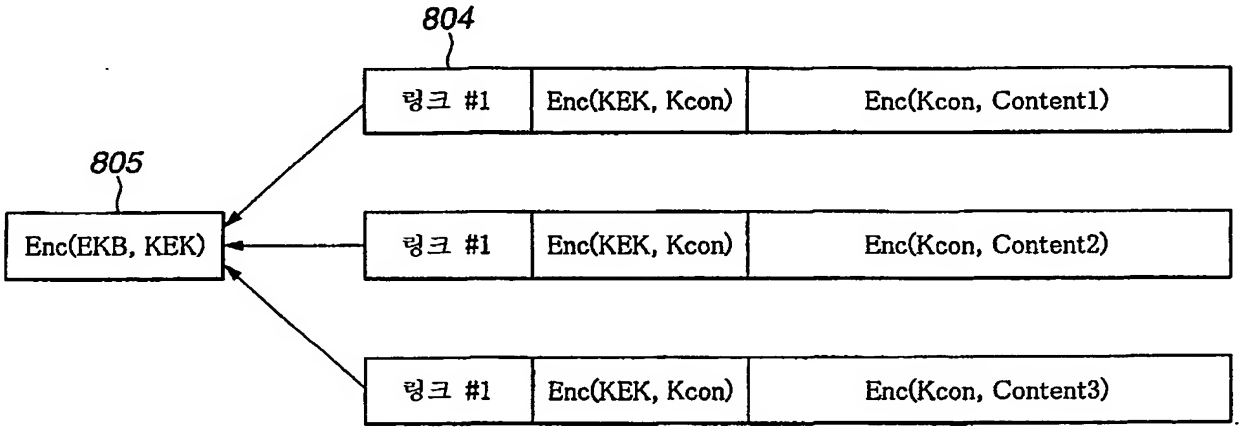
도면 7


(A)
K0000
K0001
K0010
K0011
K000
K001
K00
K0
KR
0
1
2
3
(B)
리프 노드 어드레스 : KR
데이터(암호화 키)
태그
Enc(K(t)0, K(t)R)
{0, 1}
Enc(K(t)00, K(t)0)
{0, 0}
Enc(K000, K(t)00)
{1, 1}
Enc(K(t)001, K(t)00)
{0, 1}
Enc(K0010, K(t)001)
{1, 1}
{L 태그, R 태그}
좌(L), 우(R) 각각의 방향으로 데이터가 있으면 0, 없으면 1
(C)
데이터 : Enc(K(t)0, K(t)R), Enc(K(t)00, K(t)0), ...
태그 : {0, 1}, {0, 0}, {1, 1}, ...

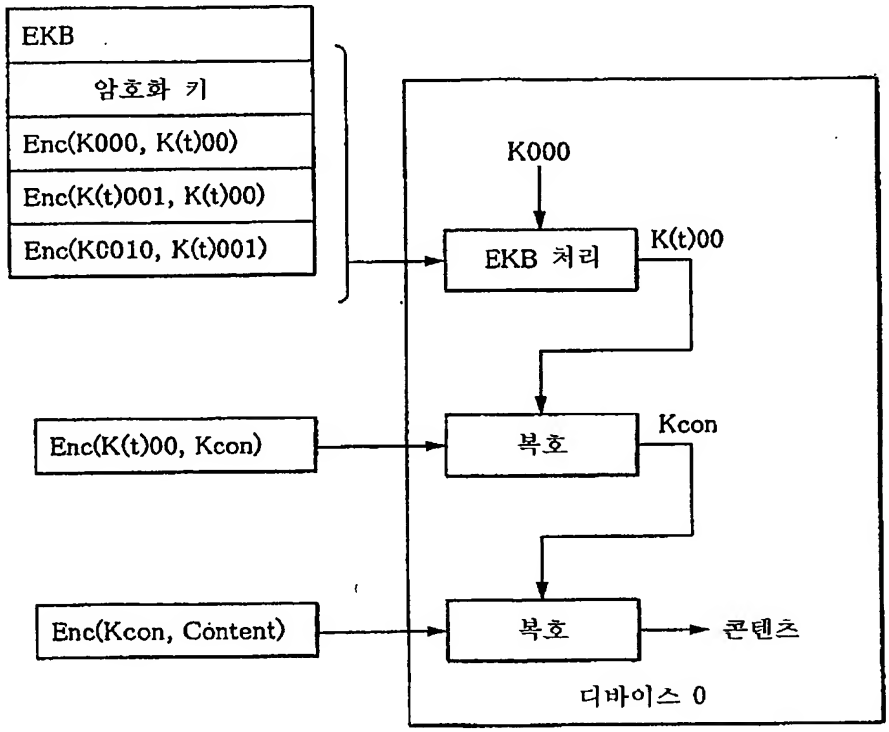
도면 8A



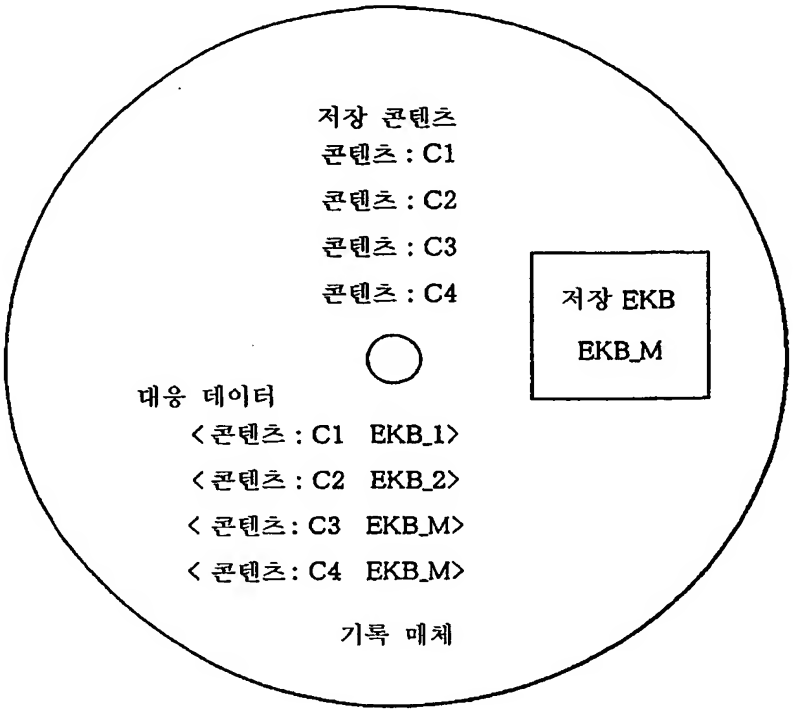
도면 8B



도면 9



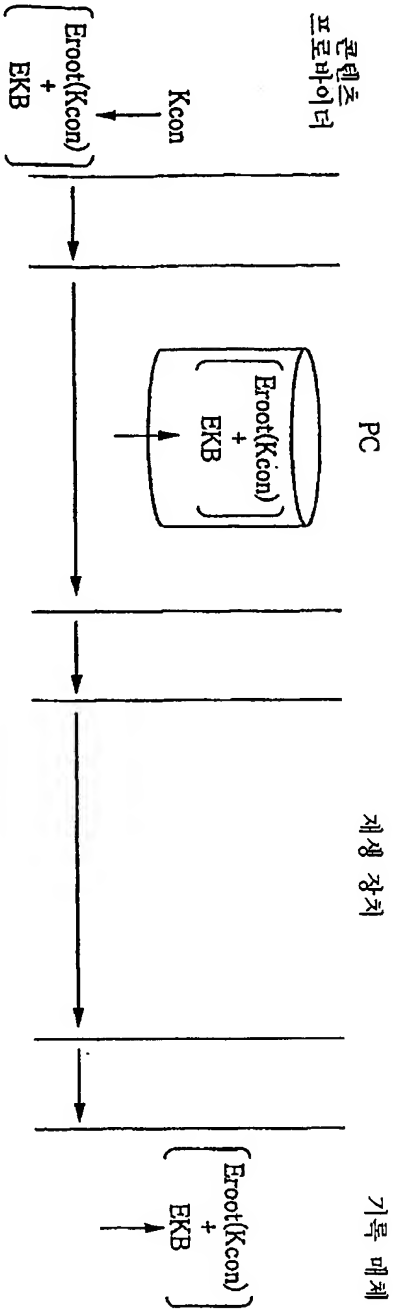
도면 10


저장 콘텐츠
콘텐츠 : C1
콘텐츠 : C2
콘텐츠 : C3
콘텐츠 : C4
저장 EKB
EKB_M
대응 데이터
<콘텐츠 : C1 EKB_1>
<콘텐츠 : C2 EKB_2>
<콘텐츠 : C3 EKB_M>
<콘텐츠 : C4 EKB_M>
기록 매체

도면 11A

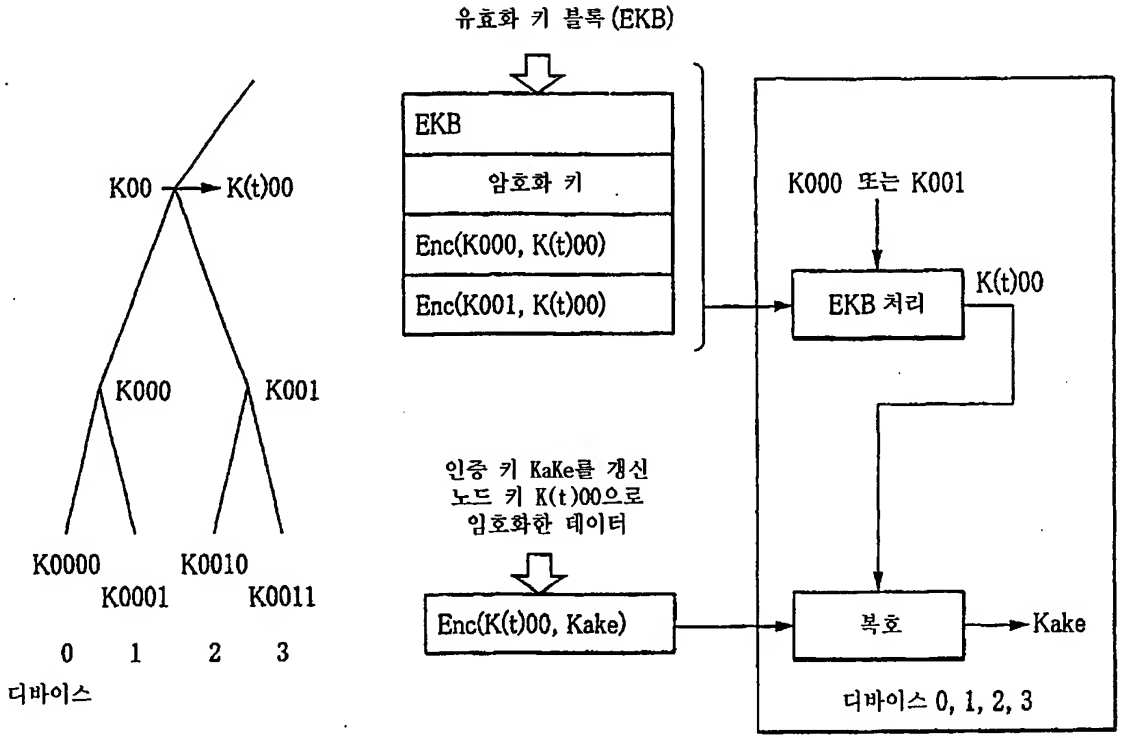
콘텐츠
프로바이더
PC
재생 장치
기록 매체
1101
AKE
AKE
AKE
Kcon
Eses(Kcon)
Estr(Kcon)
Kcon
Eses
Eses(Kcon)
Eses'(Kcon)
Eses'(Kcon)
Eses''(Kcon)
Estr'(Kcon)
Eses''(Kcon)

도면 11B

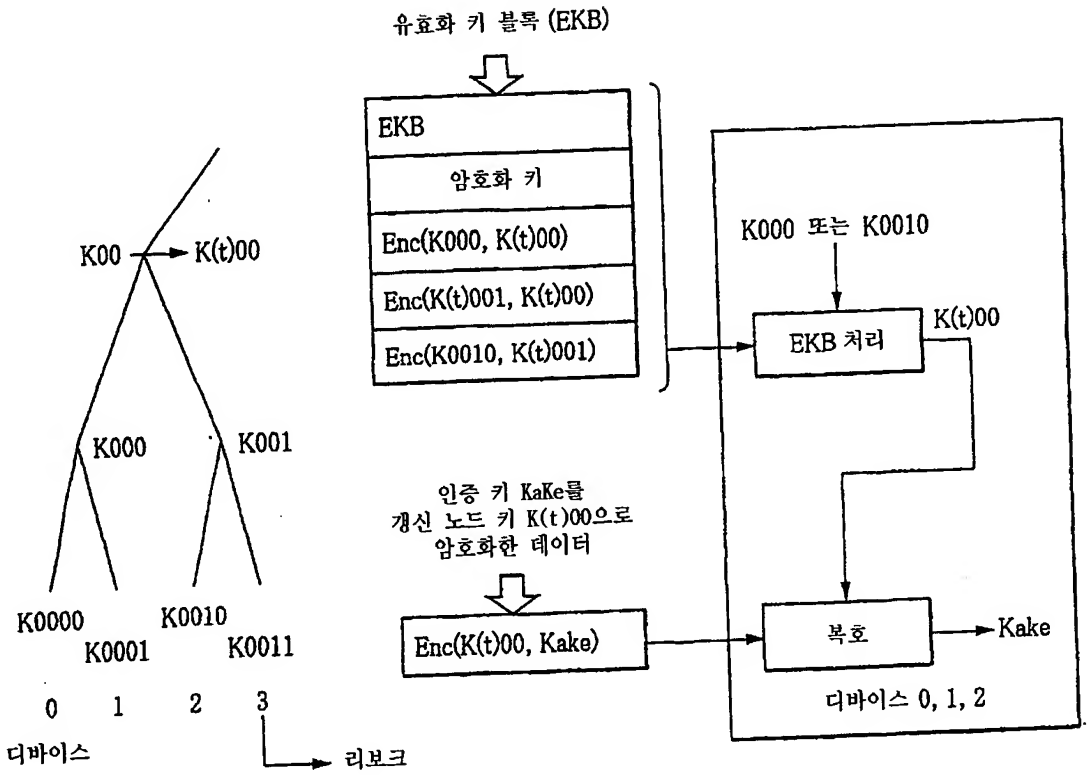

콘텐츠
프로바이더
Kcon
Eroot(Kcon)
+
EKB
PC
Eroot(Kcon)
+
EKB
재생 장치
기록 매체
Eroot(Kcon)
+
EKB

도면 12

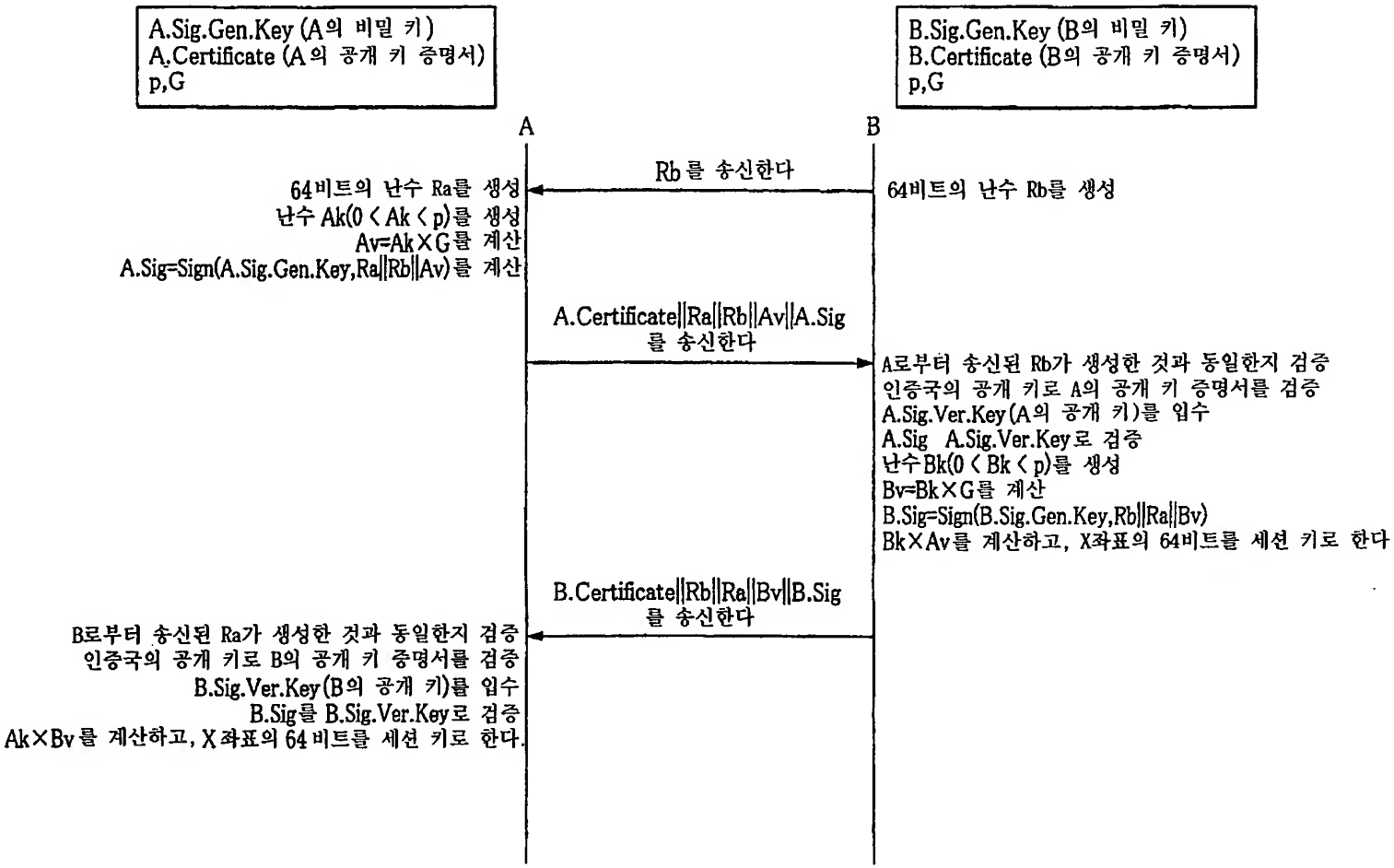
Kab
Kab
A
B
Rb||ID(b)를 송신한다
Token-AB를 송신한다
Token-BA를 송신한다
Kses를 세션 키로 한다
64 비트의 난수 Ra를 생성
Token-AB=DES(Kab,Ra||Rb||ID(b))를 계산한다
64 비트의 난수 Rb를 생성
Token-AB를 Kab를 이용하여 복호
Rb와 ID(b)가 정당한지 검증
64 비트의 난수 Kses를 생성
Token-BA=DES(Kab,Rb||Ra||Kses)를 계산한다
Token-BA를 Kab를 이용하여 복호
Rb와 Ra가 정당한지 검증


유효화 키 블록 (EKB)
EKB
암호화 키
Enc(K000, K(t)00)
Enc(K001, K(t)00)
K00
K(t)00
K000
K001
K0000
K0001
K0010
K0011
0
1
2
3
디바이스
K000 또는 K001
EKB 처리
K(t)00
인증 키 KaKe를 갱신
노드 키 K(t)00으로
암호화한 데이터
Enc(K(t)00, Kake)
복호
Kake
디바이스 0, 1, 2, 3

도면 13


유효화 키 블록 (EKB)
EKB
암호화 키
Enc(K000, K(t)00)
Enc(K(t)001, K(t)00)
Enc(K0010, K(t)001)
K00
K(t)00
K000
K001
K0000
K0001
K0010
K0011
0
1
2
3
디바이스
리보크
K000 또는 K0010
EKB 처리
K(t)00
인증 키 KaKe를
갱신 노드 키 K(t)00으로
암호화한 데이터
Enc(K(t)00, Kake)
복호
Kake
디바이스 0, 1, 2

도면 14


A.Sig.Gen.Key (A의 비밀 키)
A.Certificate (A의 공개 키 증명서)
p,G
B.Sig.Gen.Key (B의 비밀 키)
B.Certificate (B의 공개 키 증명서)
p,G
A
B
Rb를 송신한다
64비트의 난수 Ra를 생성
난수 Ak(0 < Ak < p)를 생성
Av=Ak×G를 계산
A.Sig=Sign(A.Sig.Gen.Key,Ra||Rb||Av)를 계산
64비트의 난수 Rb를 생성
A.Certificate||Ra||Rb||Av||A.Sig
를 송신한다
A로부터 송신된 Rb가 생성한 것과 동일한지 검증
인증국의 공개 키로 A의 공개 키 증명서를 검증
A.Sig.Ver.Key(A의 공개 키)를 입수
A.Sig A.Sig.Ver.Key로 검증
난수 Bk(0 < Bk < p)를 생성
Bv=Bk×G를 계산
B.Sig=Sign(B.Sig.Gen.Key,Rb||Ra||Bv)
Bk×Av를 계산하고, X좌표의 64비트를 세션 키로 한다
B.Certificate||Rb||Ra||Bv||B.Sig
를 송신한다
B로부터 송신된 Ra가 생성한 것과 동일한지 검증
인증국의 공개 키로 B의 공개 키 증명서를 검증
B.Sig.Ver.Key(B의 공개 키)를 입수
B.Sig를 B.Sig.Ver.Key로 검증
Ak×Bv를 계산하고, X좌표의 64비트를 세션 키로 한다.

도면 15

도면 16

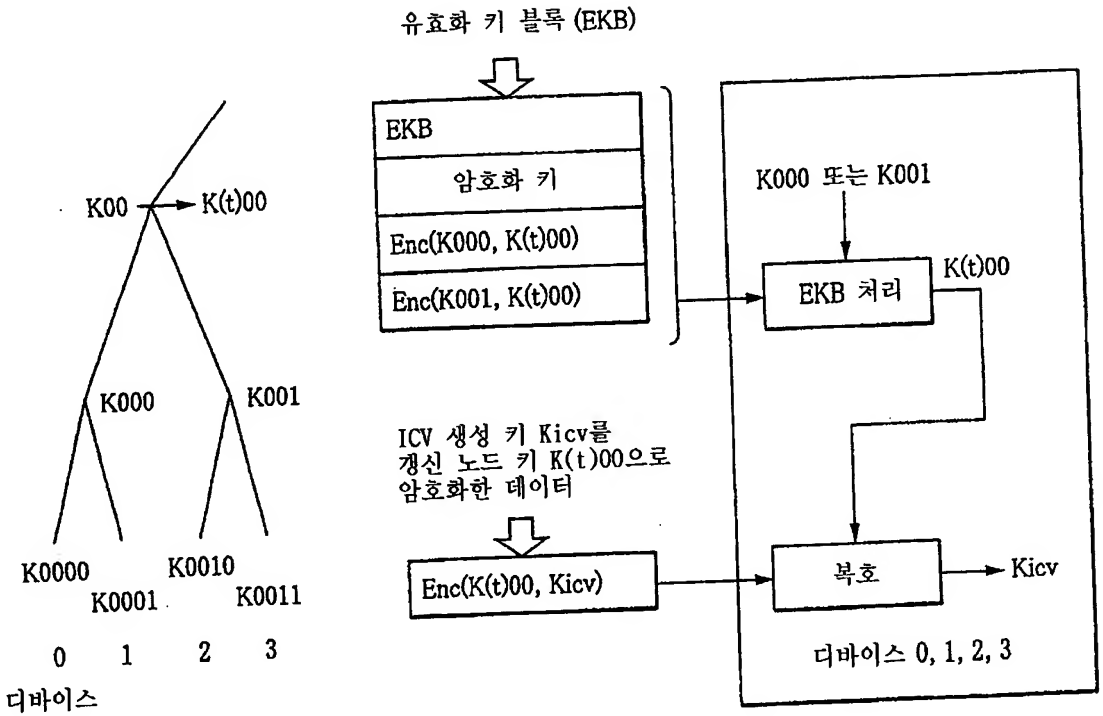
콘텐츠 프로바이더
Kcon
E(Kcon) + EKB
Ekes
E(Kcon) + EKB
AKE
PC
Eroot(Kcon) + EKB
Ekes
Eroot(Kcon) + EKB
Ekes
Eroot(Kcon) + EKB
Knode
Kleaf
Kcon
재생 장치 기록 매체

도면 17

디바이스 A
MOV $1, 0
ADD $1, 1
LOAD $1, Key1
LOAD $2, Key2
MOV $1, 1
INT 3
1701
EKB에 의한
암호화
!K)$[*DaB
&$%(Ka/?1
K9)3[Ja]a
비밀로 하고자 하는
프로그램 코드 또는 데이터
계산 속행
1702
디바이스 B
복호
실행
결과

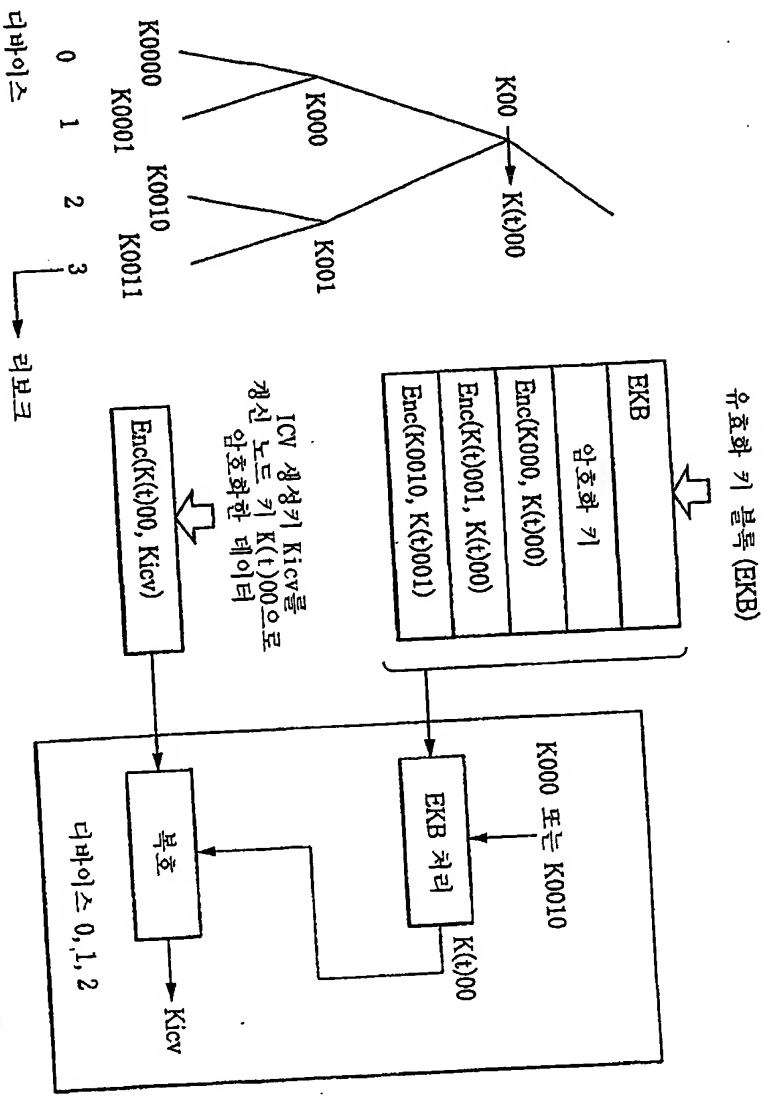
도면 18

초기치
(Initial Value : IV)
키 (K1)
메세지 M1
(처음 8바이트)
메세지 M2
(다음 8바이트)
메세지 M3
(다음 8바이트)
.....
메세지 MN
(마지막 8바이트)
I1
I2
I3
IN
DES
암호화
DES
DES
DES
암호화
E1
E2
E3
EN
메세지 인증 부호
(Message Authentication Code : MAC)

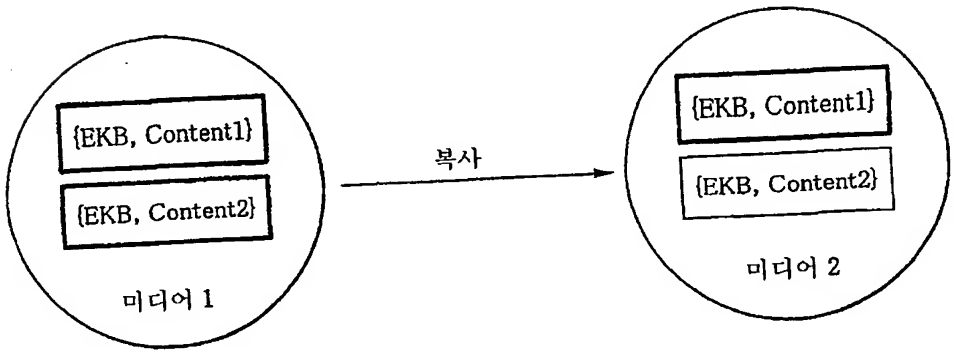

유효화 키 블록 (EKB)
EKB
암호화 키
Enc(K000, K(t)00)
Enc(K001, K(t)00)
K000 또는 K001
EKB 처리
K(t)00
ICV 생성 키 Kicv를
갱신 노드 키 K(t)00으로
암호화한 데이터
Enc(K(t)00, Kicv)
복호
Kicv
디바이스 0, 1, 2, 3
K00
K(t)00
K000
K001
K0000
K0001
K0010
K0011
0
1
2
3
디바이스

도면 19

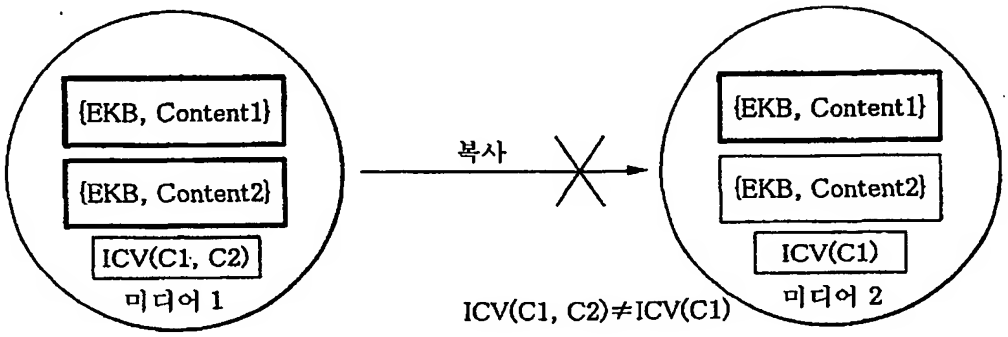
도면 20


디바이스
0
1
2
3
리보크
K0000
K0001
K0010
K0011
K000
K001
K00
K(t)00
EKB
암호화 키
Enc(K000, K(t)00)
Enc(K(t)001, K(t)00)
Enc(K0010, K(t)001)
유효화 키 블록 (EKB)
ICV 생성키 Kicv를
갱신 노드 키 K(t)00으로
암호화한 데이터
Enc(K(t)00, Kicv)
K000 또는 K0010
EKB 처리
K(t)00
복호
Kicv
디바이스 0,1,2

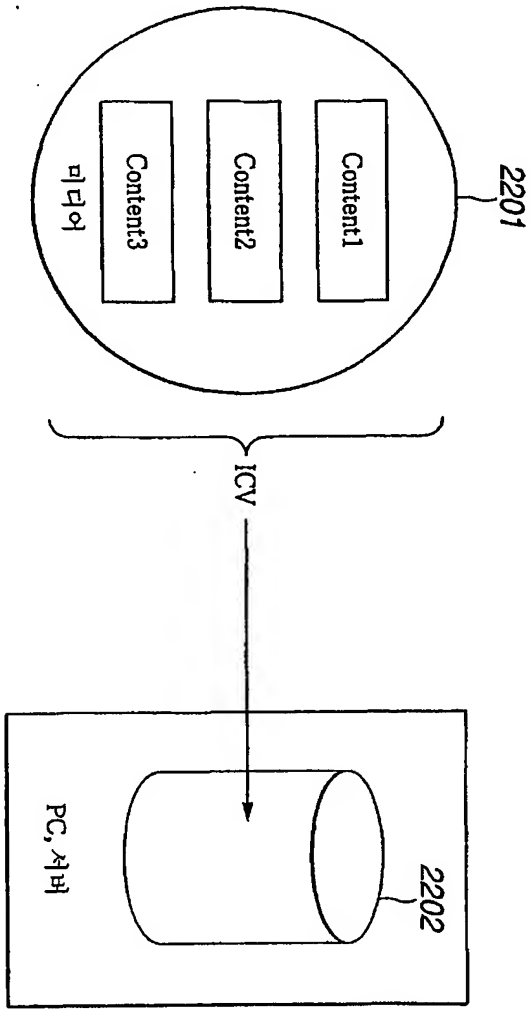
도면 21A


{EKB, Content1}
{EKB, Content2}
미디어 1
복사
{EKB, Content1}
{EKB, Content2}
미디어 2

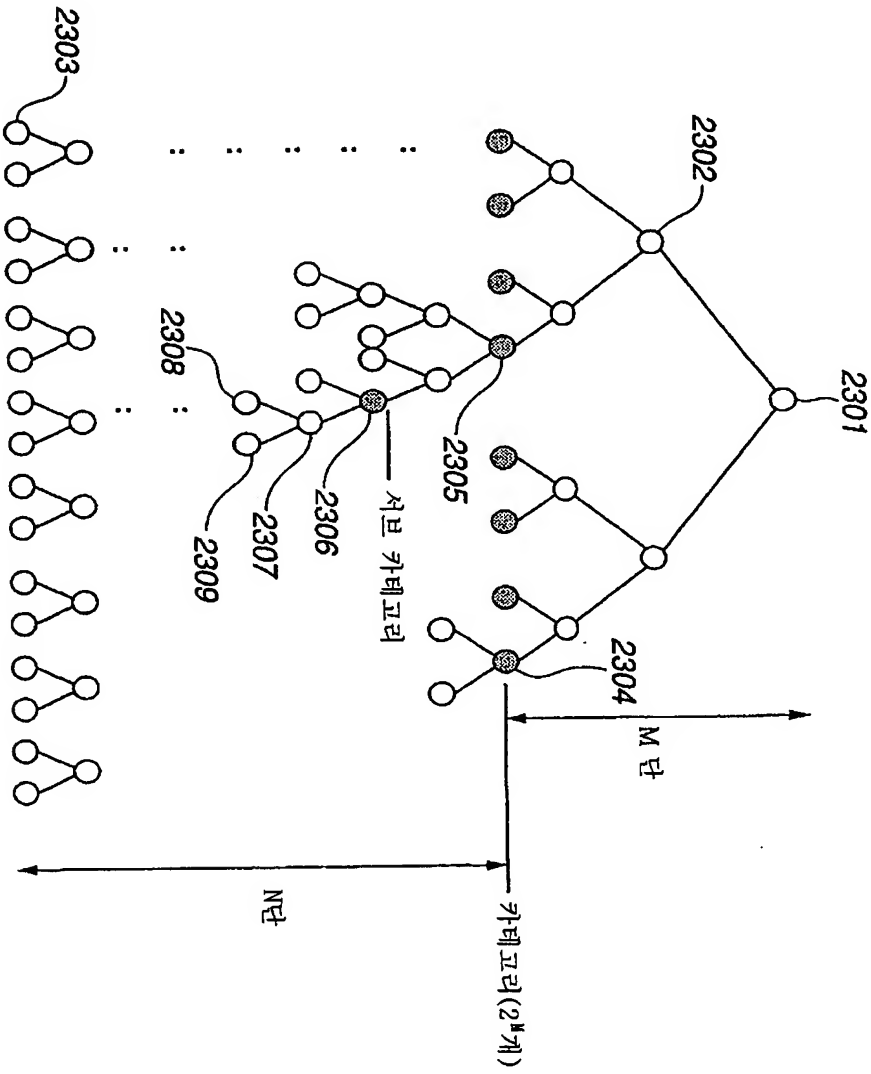
도면 21B


{EKB, Content1}
{EKB, Content2}
ICV(C1, C2)
미디어 1
복사
{EKB, Content1}
{EKB, Content2}
ICV(C1)
미디어 2
ICV(C1, C2)≠ICV(C1)

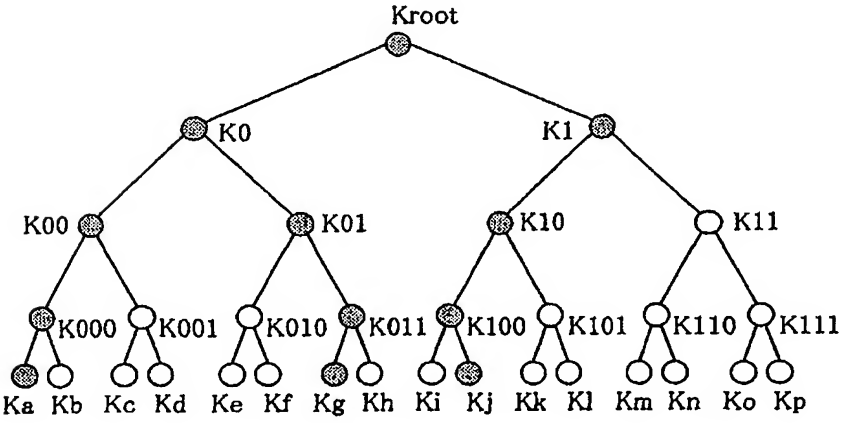
도면 22


미디어
Content3
Content2
Content1
2201
ICV
PC, 서버
2202

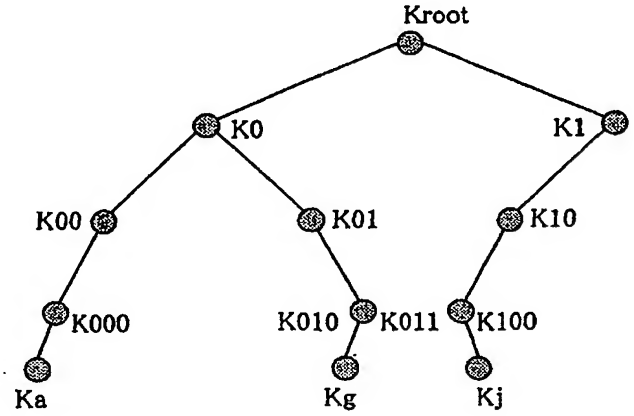
도면 23


2303
2302
2301
2308
2305
2306
2307
2309
서브 카테고리
2304
M 단
N단
카테고리(2ᴹ개)

도면 24A

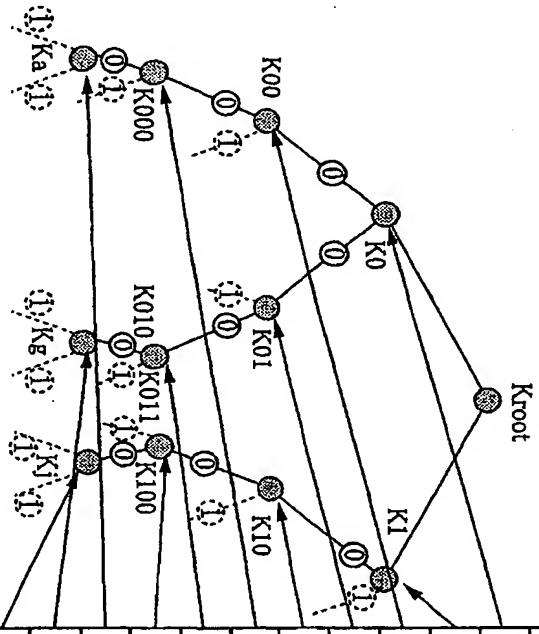

Kroot
K0
K1
K00
K01
K10
K11
K000
K001
K010
K011
K100
K101
K110
K111
Ka Kb Kc Kd Ke Kf Kg Kh Ki Kj Kk Kl Km Kn Ko Kp

도면 24B



도면 25

(A)



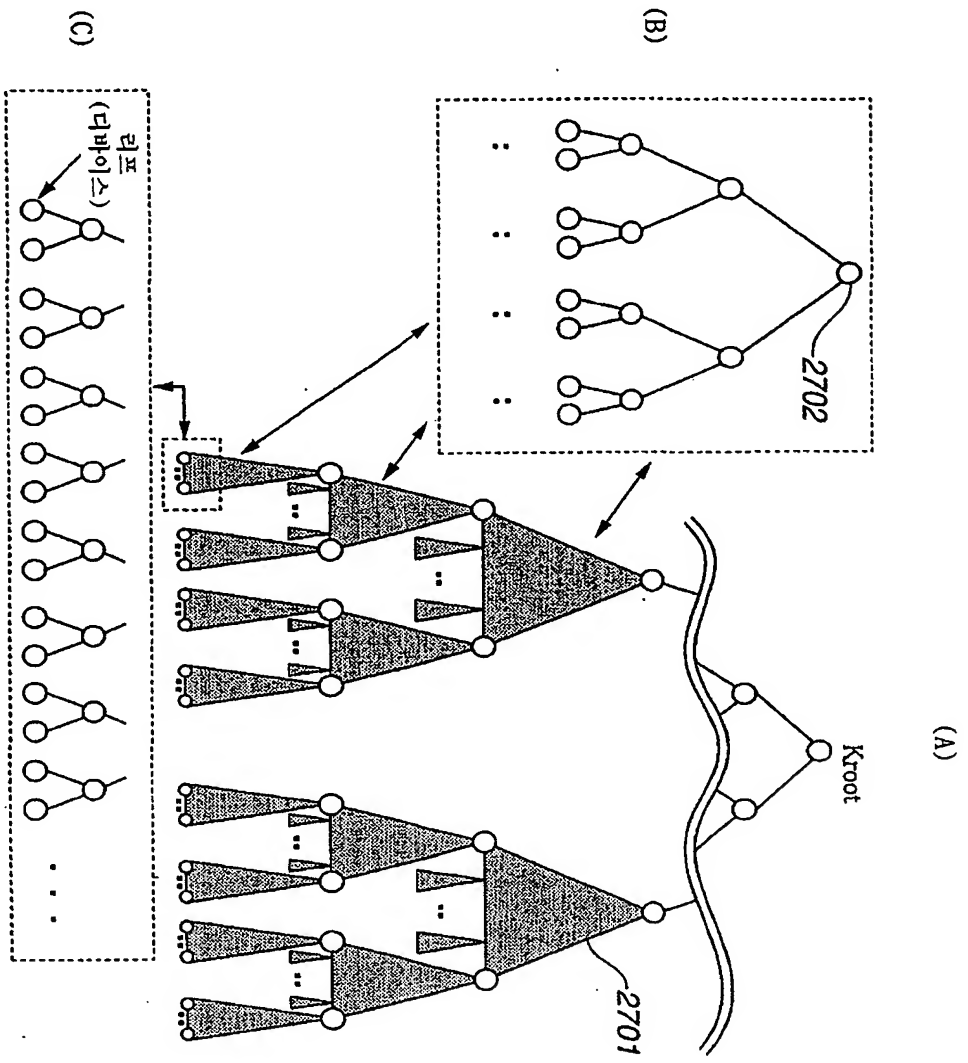
(B)

태그	데이터(암호화 키)
{0, 0}	Enc(K(t)0, K(t)root)
{0, 1}	Enc(K(t)1, K(t)root)
{0, 1}	Enc(K(t)00, K(t)0)
{1, 0}	Enc(K(t)01, K(t)0)
{0, 1}	Enc(K(t)10, K(t)1)
{0, 1}	Enc(K(t)000, K(t)00)
{0, 1}	Enc(K(t)011, K(t)01)
{1, 0}	Enc(K(t)100, K(t)10)
{1, 1}	Enc(Ka, K(t)000)
{1, 1}	Enc(Kg, K(t)011)
{1, 1}	Enc(Kj, K(t)100)
Enc(K(t)root, K(t)con)	

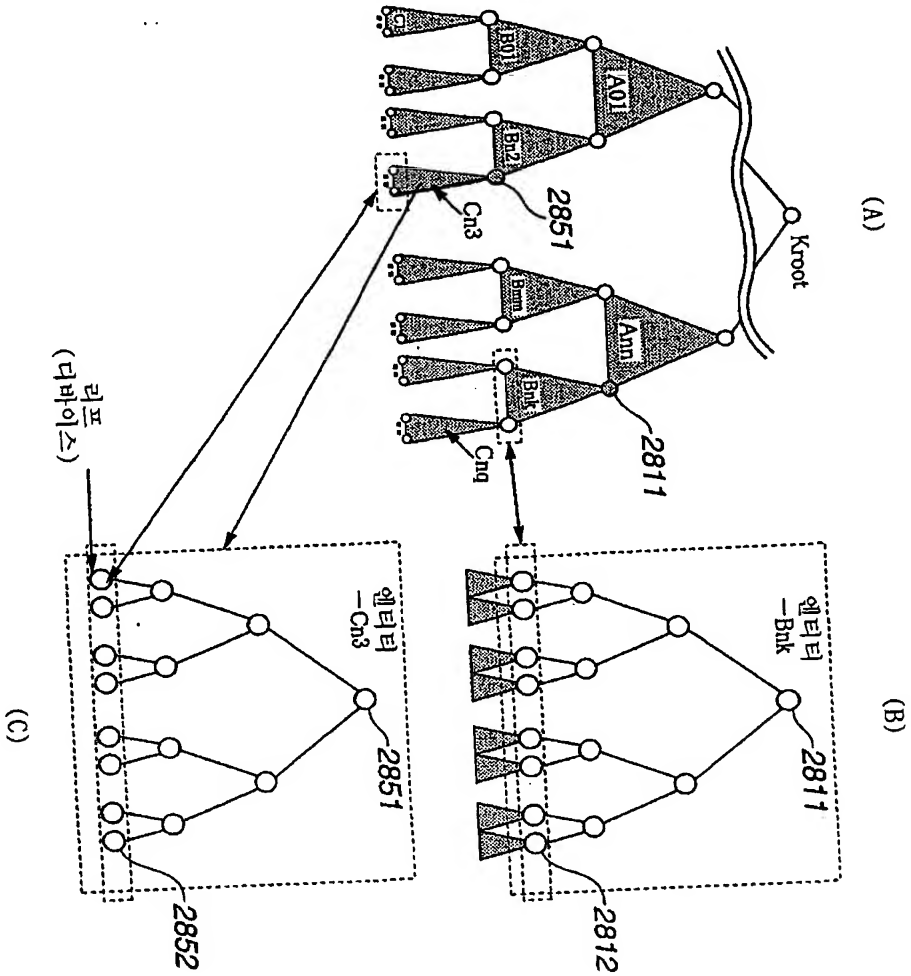
도면 26

태그	데이터(암호화 키)
{0, 0, 1}	Enc(K(t)0, K(t)root)
{0, 1, 0}	-
{0, 1, 0}	-
{1, 0, 0}	-
{0, 1, 0}	-
{0, 1, 0}	-
{0, 1, 0}	-
{1, 0, 0}	-
{1, 1, 1}	Enc(Ka, K(t)0)
{1, 1, 1}	Enc(Kg, K(t)0)
{1, 1, 1}	Enc(Kj, K(t)root)
Enc(K(t)root, K(t)con)	

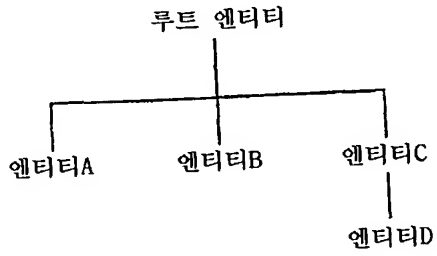
도면 27



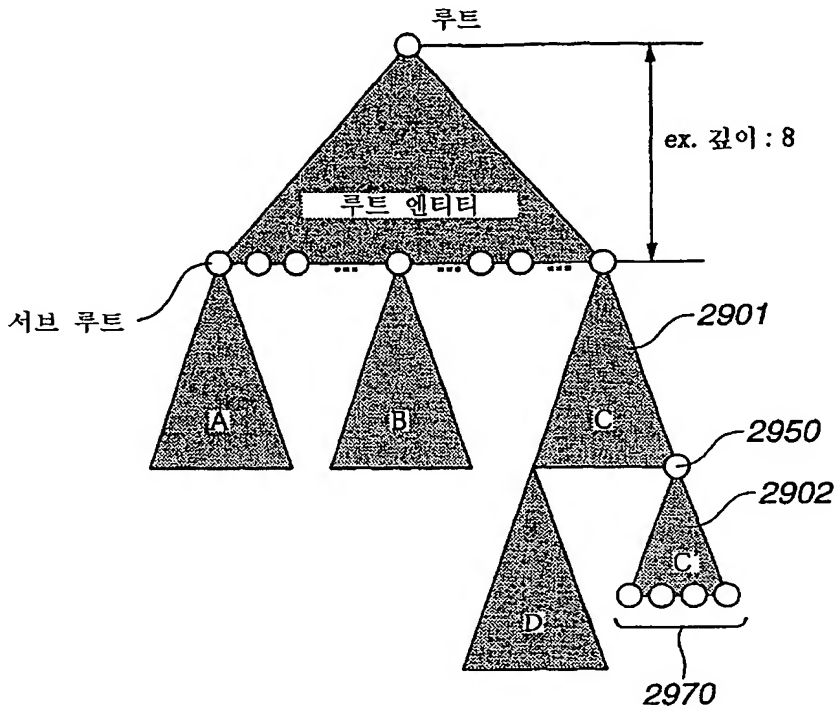
도면 28


(A)
Kroot
A01
B01
Bn2
Cn3
2851
Ann
Bmm
Bnk
Cnq
2811
(B)
엔티티
-Bnk
2811
2812
(C)
리프
(디바이스)
엔티티
-Cn3
2851
2852

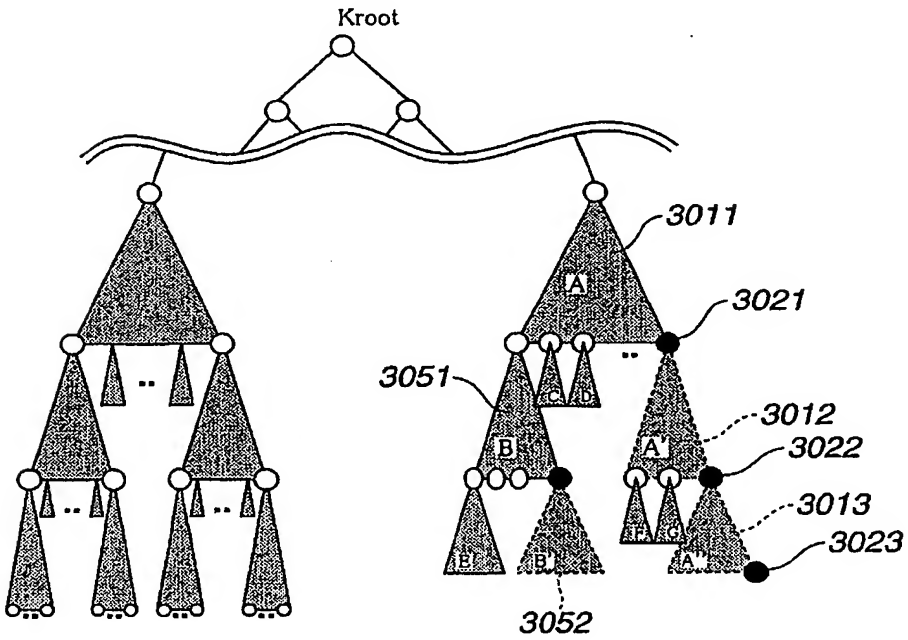
도면 29A


루트 엔티티
엔티티A
엔티티B
엔티티C
엔티티D

도면 29B


루트
ex. 깊이 : 8
루트 엔티티
서브 루트
2901
A
B
C
2950
2902
C'
D
2970

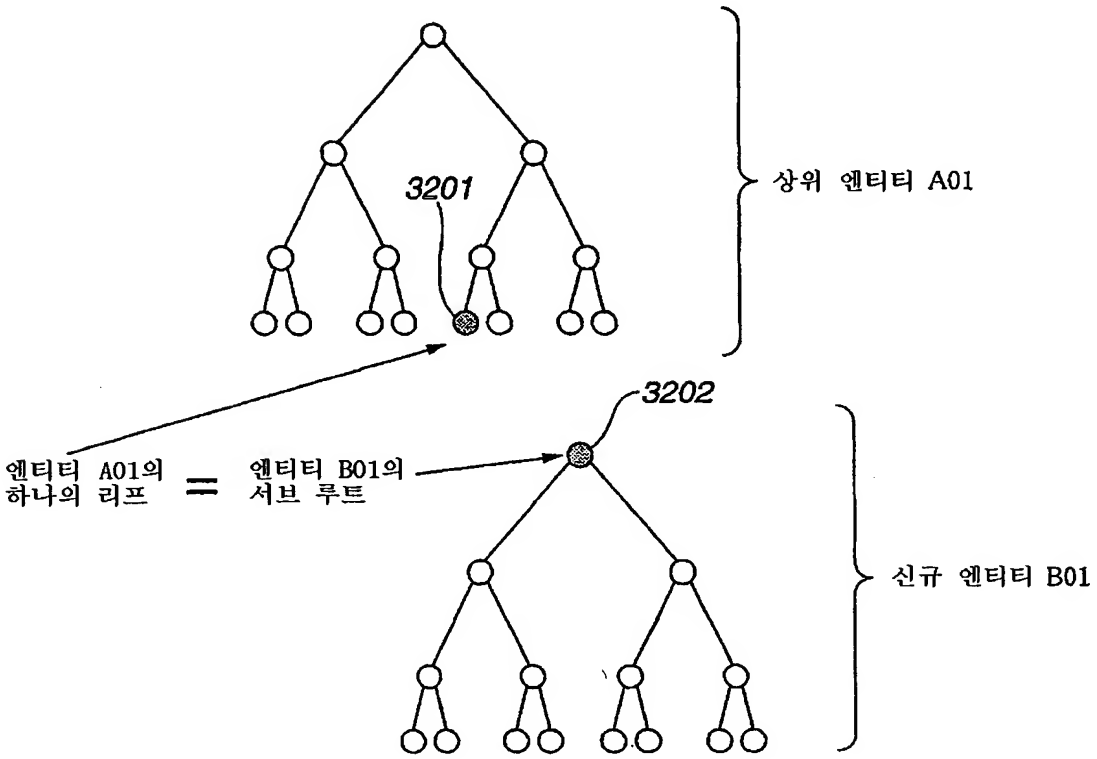
도면 30


Kroot
3011
A
3021
3051
C
D
B
A'
3012
3022
3013
3023
B'
B'
A'
3052

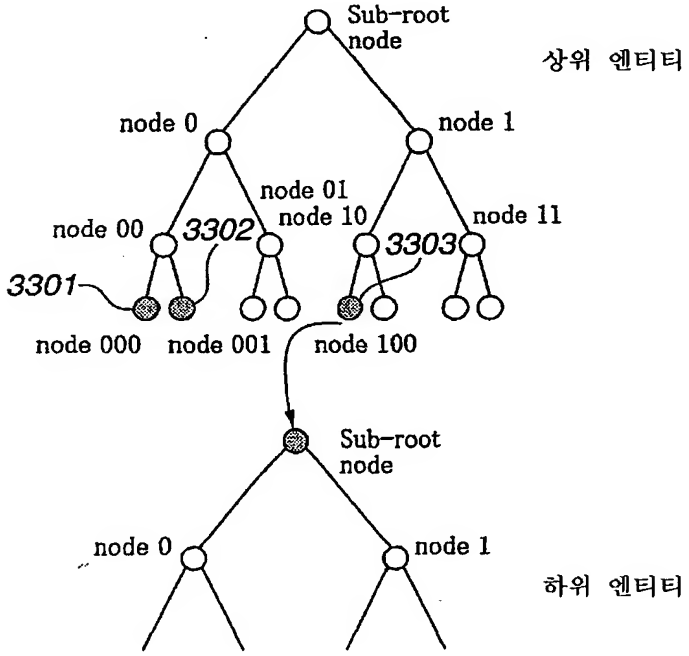
도면 31

신규(종)
엔터티(N-En)
상위(주)
엔터티(P-En)
CA
KDC
신규 등록 요구
(N-En의 공개 키를 송부)
N-En의 공개 키를 전송
N-En의 공개 키 증명서
N-En의 노드 키 송신
N-En의 엔터티 내
트리 구조를 생성
N-En의 엔터티 내
서브 EKB를 생성
N-En을 추가한
P-En의 엔터티 내의
서브 EKB를 생성
N-En의 엔터티의 서브 EKB의 송신
P-En 및 N-En 엔터티의 서브 EKB를 송신

도면 32



도면 33A



도면 33B

상위 엔티티 서브 EKB

Enc(node0, sub-root), Enc(node1, sub-root) Enc(node00, node0), Enc(node10, node1) Enc(node000, node00), Enc(node001, node00) Enc(node100, node10)

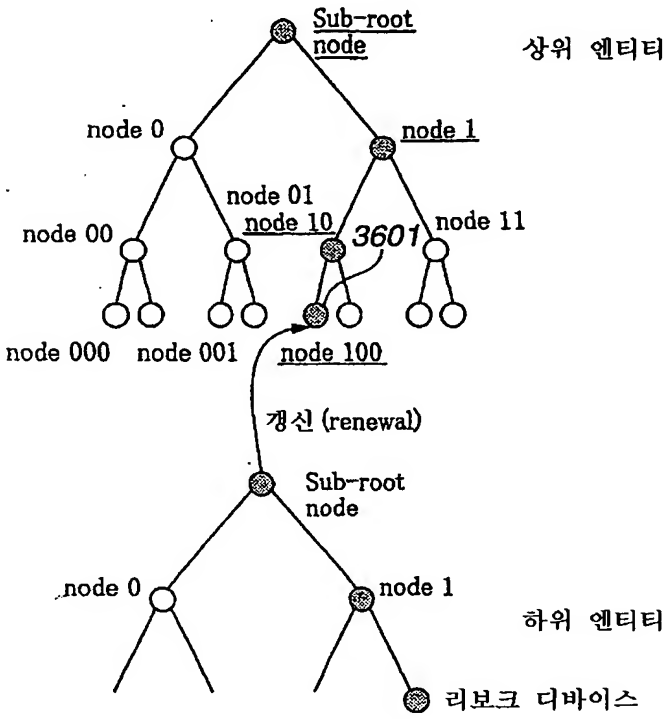
도면 34

(A)
Kroot
3411
3410
3421
3420
3431
3430
A01
B01
Bn2
Ann
Bnk
Bnm
(D)
엔터티
-Ann:3410
3411
3414
3415
3421
3420
(C)
엔터티
-Bnk:3420
3421
3424
3415
3431
3430
(B))
엔터티
-Cn:3430
3431
3434
3435
3432
리프
(디바이스)

도면 35

디바이스 관리
엔티티(D-En)
상위(주)
엔티티(P1-En)
.....
상위(주)
엔티티(Pn-En)
D-En 내의 디바이스
리보크에 의해 갱신한
갱신 서브 EKB(D)를 생성
갱신 서브 EKB(D) 송신
서브 EKB(D)의
서브 루트를 반영시켜서
갱신한 갱신 서브
EKB(P1)를 생성
갱신 서브EKB(D), EKB(P1) 송신
갱신 서브 EKB(D), EKB(P1), EKB(P2) 송신
KDC
갱신 서브 EKB(D)~EKB(Pn) 송신

도면 36A



도면 36B

상위 엔티티-갱신 서브 EKB

Enc(node0, sub-root'), Enc(node1', sub-root') Enc(node00, node0), Enc(node10', node1') Enc(node000, node00), Enc(node001, node00) Enc(node100', node10')

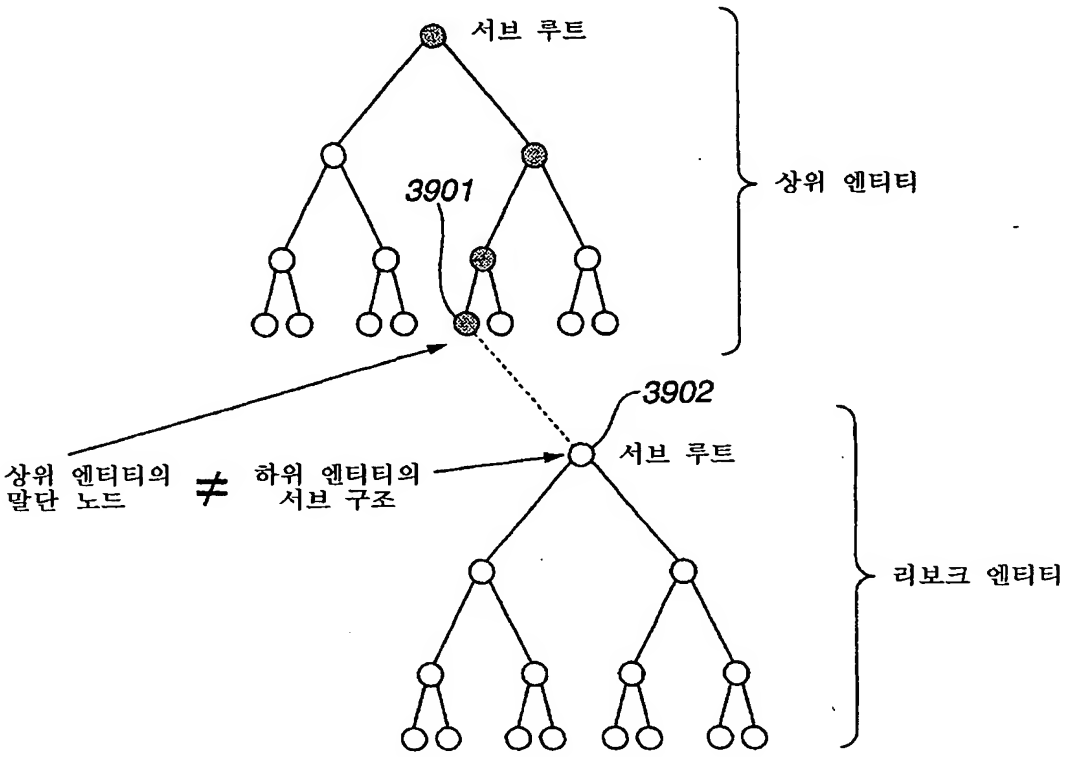
도면 37

(A)
Kroot
3711
3710
3721
3720
3731
3730
Ann
Bnk
Bnm
A01
B01
Bn2
(B)
엔터티
—Cn:3730
엔터티
서브 루트
3731
리프
(더바이스)
(C)
엔터티
—Bnk:3720
3721
3724
3715
3731
3730
(D)
엔터티
—Ann:3710
3711
3714
3715
3721
3720

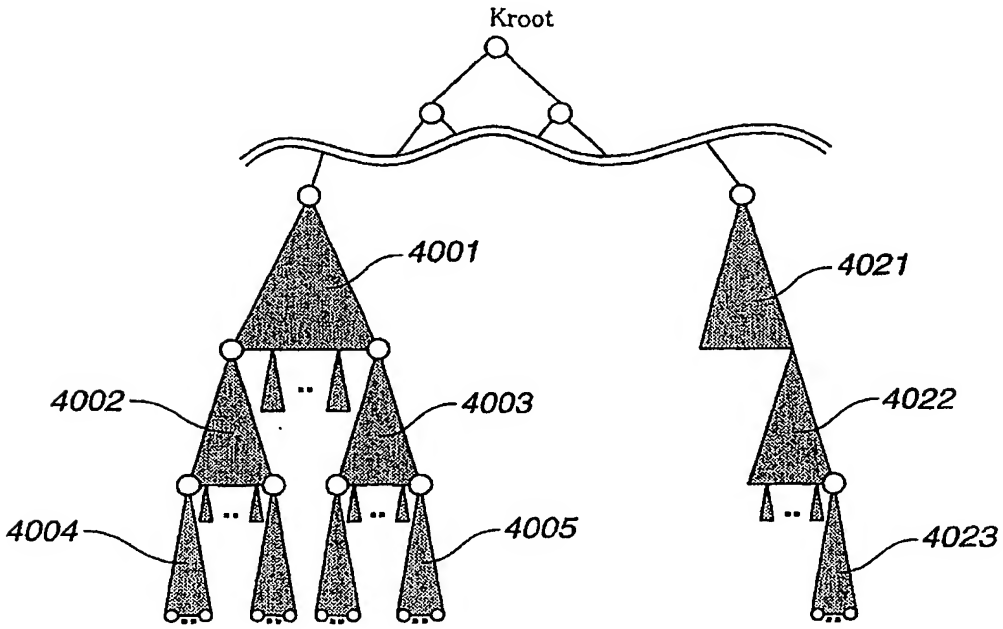
도면 38

엔터티 관리
엔터티(E-En)
E-En 내의 엔터티
리보크에 의해 갱신한
갱신 서브 EKB(E)를 생성
갱신 서브 EKB(E) 송신
상위(주)
엔터티(P1-En)
서브 EKB(E)의
서브 루트를 반영시켜서
갱신한 갱신 서브
EKB(P1)를 생성
상위(주)
엔터티(Pn-En)
갱신 서브 EKB(E), EKB(P1) 송신
갱신 서브 EKB(E), EKB(P1), EKB(P2) 송신
KDC
갱신 서브 EKB(E)~EKB(Pn) 송신

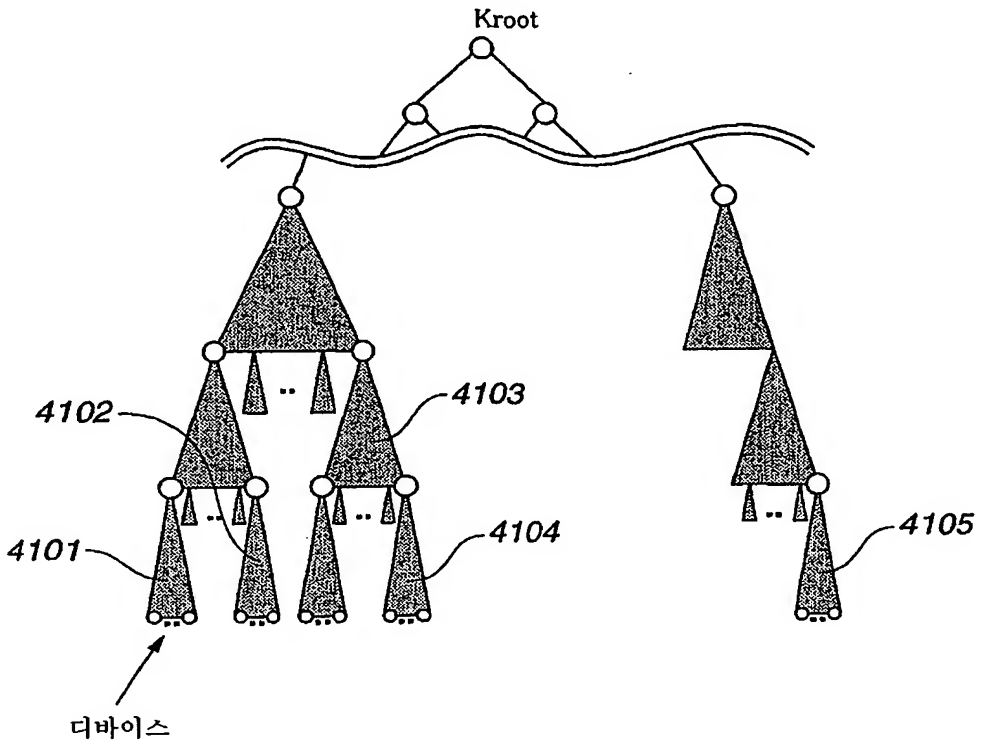
도면 39


서브 루트
3901
상위 엔티티
3902
서브 루트
상위 엔티티의
말단 노드
≠
하위 엔티티의
서브 구조
리보크 엔티티

도면 40


Kroot
4001
4021
4002
4003
4022
4004
4005
4023

도면 41



도면 42

(a)

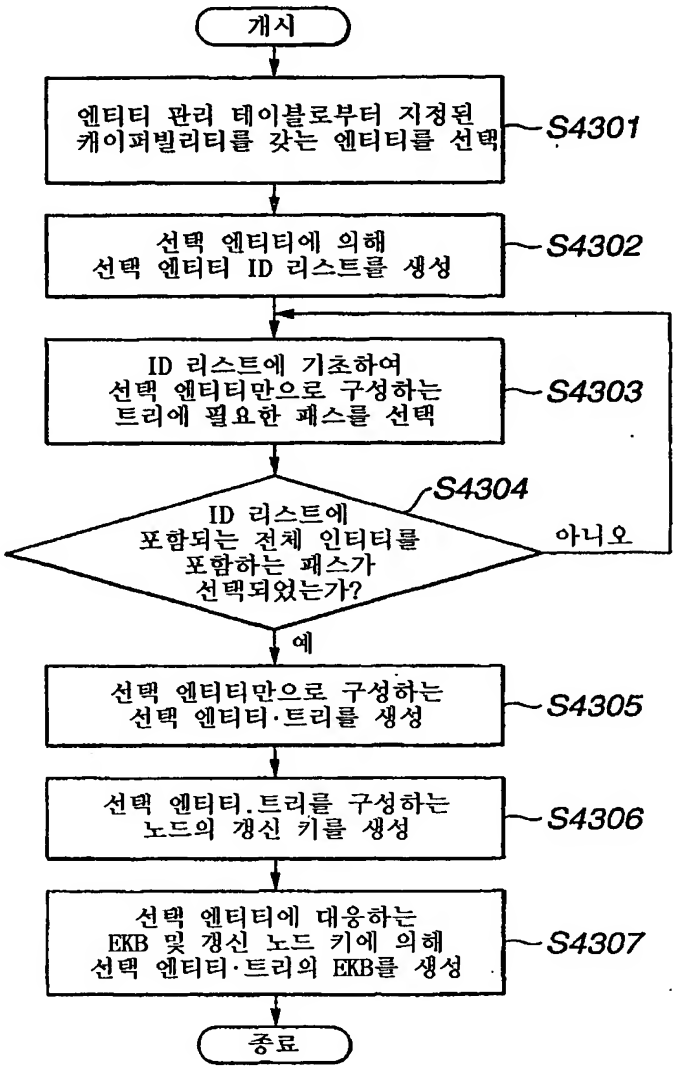
케이퍼빌리티(Capability)리스트								엔티티 ID	EKB	서브 루트
1	0	1	0	1	0	1	0	Ent0010	EKB0010	K0010
0	0	1	0	0	0	1	0	Ent0011	EKB0110	K0110
1	1	1	1	1	0	0	0	Ent0012	EKB0011	K0011
:	:	:	:	:	:	:	:	:	:	:
0	0	1	1	1	0	1	1	Ent00nn	EKB00nn	K00nn

(b)

케이퍼빌리티(Capability)리스트							
1	0	1	0	1	0	1	0

음성 데이터 재생 처리(방식A) (YES:1, NO:0)
음성 데이터 재생 처리(방식B) (YES:1, NO:0)
화상 데이터 재생 처리(방식C) (YES:1, NO:0)
화상 데이터 재생 처리(방식D) (YES:1, NO:0)
음성 데이터 기록 처리(방식M) (YES:1, NO:0)
:

도면 43


개시
엔티티 관리 테이블로부터 지정된 케이퍼빌리티를 갖는 엔티티를 선택
S4301
선택 엔티티에 의해 선택 엔티티 ID 리스트를 생성
S4302
ID 리스트에 기초하여 선택 엔티티만으로 구성하는 트리에 필요한 패스를 선택
S4303
S4304
ID 리스트에 포함되는 전체 인티티를 포함하는 패스가 선택되었는가?
아니오
예
선택 엔티티만으로 구성하는 선택 엔티티·트리를 생성
S4305
선택 엔티티.트리를 구성하는 노드의 갱신 키를 생성
S4306
선택 엔티티에 대응하는 EKB 및 갱신 노드 키에 의해 선택 엔티티·트리의 EKB를 생성
S4307
종료

신규(종)
엔티티(N-En)
상위(주)
엔티티(P-En)
CA
KDC
신규 등록 요구
(N-En의 공개 키를 송부)
N-En의 공개 키를 전송
N-En의 공개 키 증명서
N-En의 노드 키 송신
N-En의 엔티티 내
트리 구조를 생성
N-En의
엔티티 내의
서브 EKB를 생성
N-En을 추가한
P-En의 엔티티 내의
서브 EKB를 생성
N-En 엔티티의 서브 EKB를
송신함과 함께 N-En 엔티티의
케이퍼빌리티 정보를 송신
P-En 및 N-En 엔티티의 서브 EKB와 함께
N-En 엔티티의 케이퍼빌리티 정보를 송신

도면 44